

国际信息工程先进技术译丛



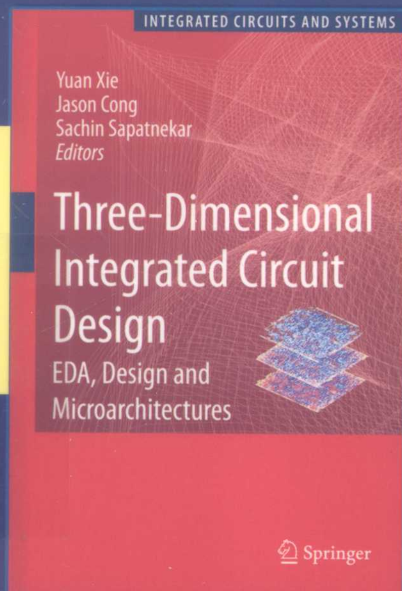
Springer

3D集成电路设计——EDA、设计和微体系结构

Three-Dimensional Integrated Circuit Design: EDA, Design and Microarchitectures

[美] 谢源(Yuan Xie) 丛京生(Jason Cong) 萨丁·斯巴肯纳(Sachin Sapatnekar) 主编

侯立刚 汪金辉 宫娜 等译



◎DAPRA微系统技术办公室Michael Fritze博士热情推荐

◎多位IEEE会士联合创作

◎3D集成电路饕餮盛宴





机械工业出版社
CHINA MACHINE PRESS

国际信息工程先进技术译丛

3D集成电路设计——EDA、设计和微体系结构

谢　源（Yuan Xie）
［美］丛京生（Jason Cong）　主编
萨丁·斯巴肯纳（Sachin Sapatnekar）
侯立刚　汪金辉　宫　娜　等译



机 械 工 业 出 版 社

图书在版编目（CIP）数据

3D 集成电路设计：EDA、设计和微体系结构/（美）谢源，（美）丛京生，（美）斯巴肯纳（Sapatnekar，S.）主编；侯立刚等译．—北京：机械工业出版社，2016.1

（国际信息工程先进技术译丛）

书名原文：Three-Dimensional Integrated Circuit Design：EDA，Design and Microarchitectures

ISBN 978-7-111-52605-6

Ⅰ．①3…　Ⅱ．①谢…②丛…③斯…④侯…　Ⅲ．①集成电路－电路设计　Ⅳ．①TN402

中国版本图书馆 CIP 数据核字（2016）第 001739 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
策划编辑：任　鑫　责任编辑：任　鑫
责任校对：肖　琳　封面设计：马精明
责任印制：乔　宇
北京中兴印刷有限公司印刷
2016 年 4 月第 1 版第 1 次印刷
169mm×239mm · 15.5 印张 · 319 千字
0001—2500 册
标准书号：ISBN 978-7-111-52605-6
定价：79.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
服务咨询热线：010-88361066	机 工 官 网：www.cmpbook.com
读者购书热线：010-68326294	机 工 官 博：weibo.com/cmp1952
010-88379203	金 书 网：www.golden-book.com
封面无防伪标均为盗版	教育服务网：www.cmpedu.com

本书全面地介绍了3D集成电路设计相关的前沿技术，章节之间有侧重也有联系。第1章首先通过处理器与存储器速度差异造成的访问速度问题，引入了3D集成电路产生的原因和存在的问题。第2章介绍了3D集成电路制造相关的基本工艺问题。针对3D集成电路远比平面集成电路严重的散热问题，在第3章总结了相关的热分析和电源传输设计方法，简述了解决相关瓶颈问题的方案。随后，本书走向设计层面，在第4章介绍了带有2D块和3D块的3D布局规划算法。在第5章介绍了几种基于热分析的3D全局布局技术，并通过实验结果比较了多种3D布局技术。第6章针对的是3D集成电路的布线，介绍了基于热分析的3D布线和热通孔插入技术。第7章介绍了重排传统的2D微处理器模块的方法，对不同设计技术、方法进行了讨论。接下来，本书继续提升设计层次，在第8章讨论了3D NoC的设计，包括多种网络拓扑结构和3D片上路由器设计。第9章介绍了高能效服务器设计的3D架构研究。最后，第10章对3D集成电路技术潜在的成本优势进行了系统级分析与设计探索。

本书可作为3D集成电路相关研究人员、工程技术人员的参考用书，也可作为高等院校相关专业高年级本科生和研究生的教材和参考书。

译 者 序

本书的作者都是3D集成电路研究领域的专家，Yuan Xie教授就职于加利福尼亚大学圣巴巴拉分校（University of California at Santa Barbara）。由于他在3D集成电路架构和设计自动化上的突出贡献，在2015年获选美国电气与电子工程师协会会士（IEEE Fellow）。Jason Cong教授现为加利福尼亚大学洛杉矶分校（University of California，Los Angeles，UCLA）计算机系教授，系主任，北京大学客座教授。他于2001年获选美国电气与电子工程师协会会士（IEEE Fellow）。Sachin Sapatnekar教授在明尼苏达大学就职，曾任IEEE transaction of CAD主编，美国电气与电子工程师协会会士（IEEE Fellow）。相信本书的出版，可以把几位会士对3D集成电路设计的真知灼见带给读者。

在对集成电路发展高度重视的时代，国家要进步离不开对最新技术的关注。3D集成电路作为集成电路行业中最有潜力的新技术之一，其研究成果很值得我们学习。本书从3D集成电路的制作工艺到设计、热分析乃至服务器架构，全面地介绍了3D集成电路相关的新技术，非常难得。在感谢原作者杰出工作的同时，也必须感谢机械工业出版社慧眼拾真，为我们大家引荐了本书。

本书得以完成，是大家努力的成果。侯立刚负责翻译了其中第1、3、4、5、6、8章，汪金辉、宫娜老师翻译了第2、7章，付婧妍翻译了第9章，王中超翻译了第10章。侯立刚、付婧妍、王中超负责了全书的统稿工作。在翻译的工程中，也得到了同事和家人的大力支持，在此一并表示感谢！

另外，本书的出版过程中，还得到了机械工业出版社的大力帮助，特此表示衷心的谢意。

特别需要说明的是，本书的翻译得到了北京市教委面上项目（JC002999201401）和北京市“青年英才”计划的资助。

由于本书内容丰富，在3D集成电路领域中有很多创见，其中许多词汇并没有确切的中文与之对应，这为本书的翻译增加了不少难度。尽管译者一直认真思考，仔细求证，但难免还会存在错误疏漏，恳请广大读者批评指正。译者联系方式为houligang@ bjut. edu. cn。

侯立刚

原　书　序

我们生活在一个大变革的时代，就像摩尔定律描述的那样，电子世界最近的几十年见证了前所未有的成长和进步。摩尔定律预测晶体管的集成度每1.5～2年翻一番。与此同时，单个器件的性能同步提升，功耗同步减小，这也让集成电路的总功耗保持可控。没有趋势可以永远持续，不幸的是摩尔定律正要面对这一现状。

几年前当CMOS器件不再能继续沿着以往的规律按比例缩小时，困难就出现了。像栅氧化层厚度等关键器件参数已经无法简单地按比例缩小了。结果，器件的关断电流开始以惊人的比率飙升。这些传统的按比例缩小技术的连续问题致使集成电路时钟频率降到几GHz的范围以内。当然，芯片可以选更高的时钟频率，但那样热问题将无法控制。这引发了近期处理器的多核趋势，让每个处理器核最多工作在几GHz。这样做的目标是不再通过提升速度提高整体性能，而是通过增加更多的处理器核来增加并行处理能力以实现性能提升。存在的挑战是确保通用代码可以有效地并行化。

另一个提升CMOS工艺性能的潜在解决方案是，3D集成电路（3D IC）。通过在垂直方向实现多个有源“层”的工艺，益处显著。全局线网将大大缩短，互连带宽将大大提升，延迟将显著减小。大量的低延迟缓存将被应用，智能物理设计将协助减轻热和电源传输的热点。三维集成电路工艺为保持摩尔定律预计的进展提供了可实现的路径，而且不需要传统的按比例缩小。这是未来的一个至关重要的机会。

美国国防部先进研究项目局（DAPRA）在许多年前就意识到3D IC工艺的重要性，并对于此领域中具有军用相关性和应用潜质的项目进行了仔细的筛选和资助。这种工艺也有许多潜在的商业价值。针对智能成像器、异质3D集成堆叠和数字性能提升，DAPRA的微系统技术办公室近年启动了一系列基于3D IC的项目。本书中很多章节的研究结果是由DAPRA在3D IC领域赞助的推动下实现的。

3D集成电路技术现在尚处于早期阶段，有些工艺刚刚实现而更多的正在开发过程中。然而，它的潜力如此之大以至于一个专注的团队已经开始认真研究和3D IC相关的EDA、设计以及架构问题，这些研究均已在本书中详细总结。第1章由来自IBM公司，对设计和工艺均很擅长的专家很好地介绍了这个领域。第2章由来自IBM公司的工艺研究者提供了精彩的3D IC关键工艺总结，必将使任何设计者和架构工程师受益。第3～6章由加利福尼亚大学洛杉矶分校（UCLA）和明尼苏达大学的研究者阐述了3D IC设计自动化（EDA）的重要问题。这些章节所涉及的关键问题包括热、电气和布局的管理方法，其挑战是在建模和物理设计流程期间的

一个多层电子堆栈。第7~9章介绍了3D设计问题，包括由佐治亚理工学院的专家介绍的3D处理器设计，由宾夕法尼亚州立大学的专家介绍的3D片上网络（NoC）架构，以及由密歇根大学和英特尔公司的专家介绍的高能效服务器设计的3D架构研究。本书最后由宾夕法尼亚州立大学的专家介绍了对3D IC技术潜在成本优势的系统级分析。

像我前面提到的，我们生活在一个大变革的时代。这些变化让人惊惧，因为像摩尔定律这种长期证明的预测和规律不再有效了。然而挑战的时刻也是尝试新想法的绝佳机会。

3D集成电路技术就是这样一个新想法。本书将成为重要的先锋角色，引领研究人员和集成电路工艺进入这一全新的领域。

DARPA 微系统技术办公室
Michael Fritze 博士，于弗吉尼亚州阿灵顿，2009年3月

原书前言

对于一个观察者来说，纽约市在集成电路设计人员心目中有一个特殊的位置。曼哈顿结构，模仿了著名的街区和街道，常规应用于物理设计：在此范例下，所有的形状被拆分成矩形，每条线网都垂直或平行于其他线网。3D 集成电路的出现将这一规律扩展到曼哈顿的另一个显著特点，即它的摩天大楼，原因是 IC 向上搭建，有源器件层堆叠在一起。更准确地说，不再像传统的 2D IC 工艺仅有一个带有有源器件层的衬底和此层之上的几层互连线，3D IC 将多个芯片层堆叠在一起。这提升了硅实际空间的利用率，并使在堆叠中使用高效的通信结构（就像摩天大楼的电梯）成为可能。

从主流的 2D 规范到 3D 显然不是一小步：在不止一个方面，这种改变都增加了一个新的 IC 设计维度。三维设计需要创新的工艺和制造技术来可靠地、经济地将多层电路堆叠在一起，需要设计方法从电路层次到架构层次均能充分利用 3D 的优势，需要计算机辅助设计（CAD）技术能在设计的所有阶段分析和优化电路。在之前的几年中，随着 3D 工艺技术逐步成熟，3D 集成电路已经实现，该领域已经引起了研究关注的大爆发。本书的目的就是捕捉当前最先进的技术并向读者综合地介绍当前制造工艺、设计方法、计算机辅助设计（CAD）技术的进展。本书的一部分内容由该领域最前沿的研究构成，细致探讨了设计 3D 集成电路的挑战与机遇。

3D 集成电路的历史要向前追溯很多年，最早可追溯到几十年前日本政府资助的一个重要研究项目。只是在最近几年，3D 集成电路才吸引了关注，如今被认为是一种可实现的选项。时至今日，主要的半导体工业巨头们都在该领域投入了大量的资源。因此，3D 技术正处在一个出现重大飞跃的平衡点上。该技术的内容和动机在第 1 章中介绍。

3D 集成电路覆盖领域广，现有的多种 3D 技术为成本和性能的折中提供了广阔的空间。这包括多个芯片置于一个衬底上的硅基类工艺，中间层间距为数百微米量级的晶圆级堆叠，以及中间层间距为数十微米量级的薄芯片/晶圆堆叠。前两者的优点是可以实现紧密封装和更高层的集成，缺点是常常造成层间通信的大量性能开销。最后一种，拥有更小的层间距，不仅增加了集成层数，而且利用了能实际显著改善等效 2D 实现的新结构。这样先进的技术是本书的重点，此类技术的前沿示例在第 2 章中有详细阐述。

在构建 3D 结构的过程中，大量的问题需要 CAD 工具和设计技术解决。从 2D 到 3D 的变化基本上属于拓扑结构的变化，因此为 3D 芯片设计布局规划、布局及

布线工具是非常重要的。此外，3D 芯片单位尺寸内比对应的 2D 芯片需要更大量的电流，导致了严重的热和电源传输的瓶颈。用于 3D 的任何物理设计系统必须包含热分析部分，并且必须关注电源传输网络的构建问题。所有这些问题在第 3 ~ 6 章中着重进行了说明。

在系统层次方面，3D 架构可用于构建新结构。对于传感器芯片，传感器可以置于顶层，模拟放大器置于其下，数字信号处理电路再下一层。这种思路已经在概念和实现层次上应用于图像传感器和天线阵列了。对于处理器设计，3D 结构使存储器可以堆叠在处理器之上，从而允许两者间高速的通信，这将解决此类系统中最主要的一个性能瓶颈。第 7 ~ 9 章讨论了几个系统设计的例子。最后，第 10 章给出了一种 3D 电路成本分析的方法。

我们希望本书能带给读者当前 3D IC 设计的整体状况和该技术未来的展望。

Sachin Sapatnekar

目　　录

第1章　介　　绍

Kerry Bernstein

就像钢架构突然让摩天大楼超越了石制建筑的12层极限[6]一样，四项关键工艺的进步让20年前的梦想家（如美国的Jim Meindl和日本的Mitsumasa Koyanagi）提出的3D集成电路的概念[2]能够实现。这些要素是：（1）低温绑定；（2）层到层的移动和对齐；（3）层间的电气连接；（4）一种有效的发布工艺。这些将是建设我们新电子摩天大楼的起重机。与此同时，目前创建这样非凡的电子结构的动机仍然不甚了然。这个观点最终出现在一篇非正式杂志的文章中[5]，而其提出的具有先见之明的想法没被立刻发现。TI公司的Doug Matzke在1997年发现信号的局域性将最终限制处理器的性能和吞吐量增益。该发现以光速般在媒体中传开。很明显在当时线网延时的改进跟不上器件的改进速度，为了实现同步的性能提升，互连需要持续地引入新的材料和结构。事实上，历史已经证明了这个命题的正确性。图1-1示出了1995年以来互连问题上的压力。在图中，圆圈代表在一个周期内可访问到的区域，很明显它的半径随着时间而缩小，这意味着更少的片上资源可以在一个周期内到达。3种趋势共同导致其半径的单调递减[1]：

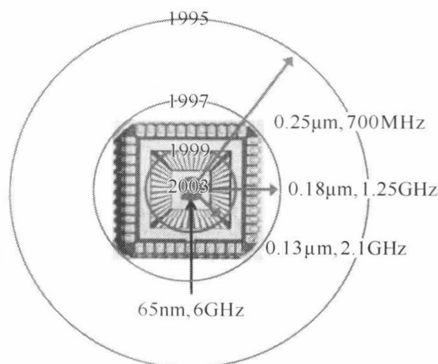


图1-1　单处理器跨芯片延迟的理想风暴图：（1）线网不按比例缩小；（2）芯片面积增长；（3）更短的FO4步骤。跨芯片延迟的功耗增大[1]

1）线网不按比例缩小。排除冶金学家和线上后端工程师的不懈努力，在最好的情况下，芯片的互连延时将在下一代芯片中保持不变。在每代芯片中都加入了新材料，比如更低介电常数的电解质、铜和更多的金属层的情况下，这很说明问题。考虑到同一时期的器件性能通过按比例缩小得以提升，可访问半径注定缩小。

2）芯片面积增长。如果每代芯片的面积都随按比例缩小的轨迹变化而相应地缩小，器件和线网延时的不平衡问题将更加严重。实际的趋势反而相反，由于计算机吞吐量的提高，其结构的改进，使得相对芯片面积一直增加。即使在媒介中以光速传送，信号仍需要更长的时间以穿过整个芯片。即使芯片面积没有增长，设计将被努力提升以达到与上一代工艺时相同的周期时间约束。

3）更短的周期。以上命题在周期时间约束没有保持固定的情况下持续减小，变复杂了。但为了更充分地利用片上资源，在每个相邻的代之间，设计师已成功地实现每周期减小相当于“4扇出反相器”的延时，使得流水线气泡将不会像在更长周期时间下一样严重地阻塞片上功能单元。在这种情况下，信号不仅需要在未改进的线网上走更远，而且需要比之前更少的到达时间。

图1-1中示意的结果说明，单处理器已经失去了在一个周期内访问整个芯片资源的能力。一种补救的方式是在单处理器中加入多个相同的资源以保证至少其中一个可在一个周期内访问到，而这让问题更加棘手。

以上示意的趋势是由工业数据实际证实的。图1-2示出了在过去10年中及之前在会议中记录和发布的处理器的面积与SpecInt2000（一种微处理器性能的测量指标）的性能比率。其趋势的外推说明该实现是存在一定极限的。

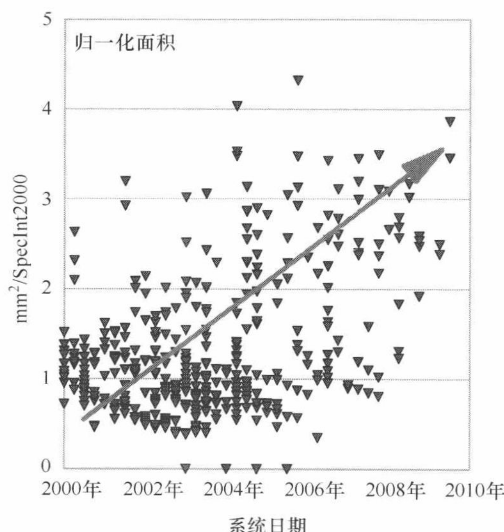


图1-2 面积与SpecInt2000性能比

尽管结构的方式可使性能提升，但这种方式的面积费用昂贵，并以跨芯片信号延迟为代价。一个费用的例子如图1-3所示。随着每周期的级（stage）数的下降（见上述第3点描述），处理器必须存储大量的中间结果，因此需要更多的锁存器和寄存器。Srinivasan展示出了对于固定的累计逻辑深度，随着每周期FO4等效延

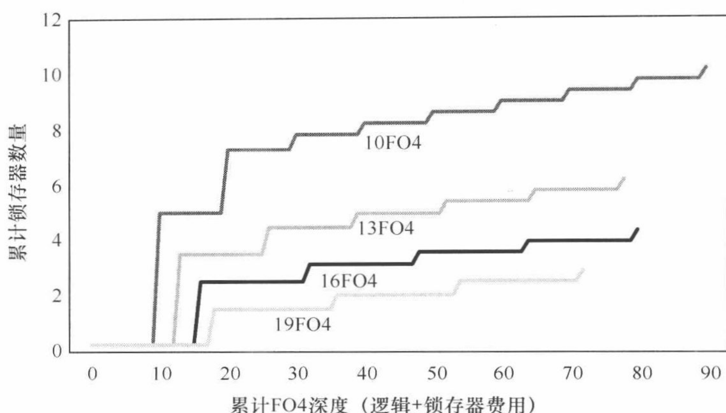


图 1-3 锁存器和流水线深度的超线性增长趋势© 2002 IEEE.
来自本章参考文献［7］，并已获得许可

时级数的下降，所需的寄存器增加[7]。增加的寄存器不仅消耗面积，而且也需要占用周期中更多的比率以满足时序边界。

尽管存在提到的以上限制，但定性地解释工业界通过工艺按比例缩小获得成功的某些原因非常合适。随着在电学上资源不断远离，微处理器结构开始青睐多核，如 SMP 机，其中每个核都是一个相对简单的处理器，仅顺序执行指令。在这些多核系统中，独立处理器核分担了它们更纠结的单处理器前辈大部分复杂度。我们可以简单讨论一下，在芯片输入输出带宽可以提供每个核所需要数据的情况下，增加的片上处理器核维持了微处理器性能改进。事实上，这在多核系统的早期是正确的。并且，恰恰在器件延时持续改善、互连性能已成为设计者的重要限制的时候，多核处理器正好主宰了高性能处理领域。这不是偶然。

如在图 1-4 中定性地示出的，多核方式将继续提供性能改善，如图中黑色线所示，直到互连带宽再次成为性能瓶颈。在这点上来说，克服带宽限制将需要根本的模式改变，超越了在 2D 设计中成功改善互连延迟的仅改变材料的方式。3D 集成正好提供了这种能力：一旦采用，这项技术将继续扩展微处理器吞吐量直到其优势饱和，如图 1-4 中上方虚线所示。若没有 3D 技术，我们将更早遭遇多核处理技术的局限，如图 1-4 中较低的虚线所示。这个局限早在 2001 年就被发现了[2]。这些文献经常被引用：人们可以“向上”或“向外”按比例规格化未来的设计，同时互连必定是设计上不可接受的局限。文献中的图 1-5 示出了将需要不现实的 90 个布线层的设计或需要保持每个宏模块 1000 万个器件以下，来保证可布线的设计。这两种解决方案都不可行。

让我们回过头审视让 3D 集成如此及时和有效的特定结构问题。我们从检查处理器用来做什么和如何组织它们使其最有效率开始。

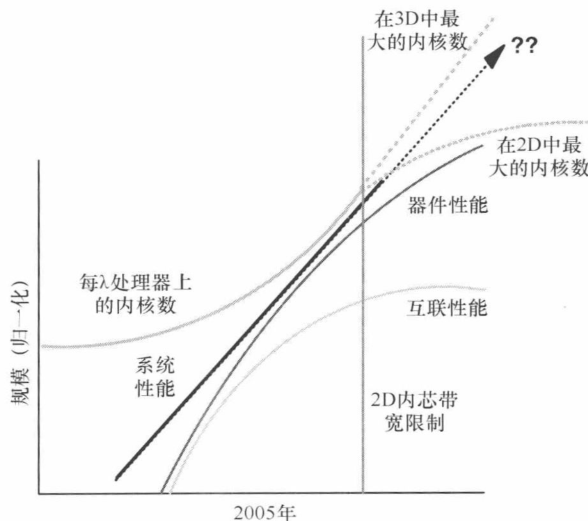
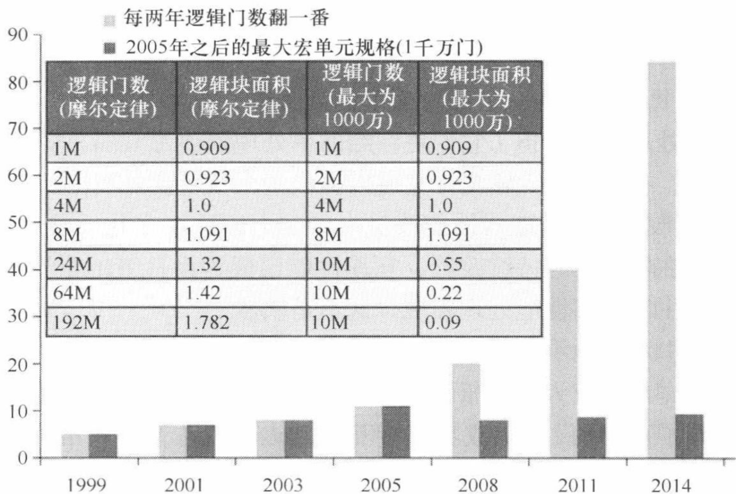


图 1-4 微处理器性能提取



逻辑门数（摩尔定律）	逻辑块面积（摩尔定律）	逻辑门数（最大为1000万）	逻辑块面积（最大为1000万）
1M	0.909	1M	0.909
2M	0.923	2M	0.923
4M	1.0	4M	1.0
8M	1.091	8M	1.091
24M	1.32	10M	0.55
64M	1.42	10M	0.22
192M	1.782	10M	0.09

图 1-5 早期 3D 预测：向上按比例规格化（“2005 年后，不可思议的布线层数。”）向外按比例规格化（“如果门/宏模块小于 10M 时可支配互连”）[2] © 2001 IEEE. 来自 Interconnect Limits on Gigascale Integration (GSI) in the 21st Century J. Davis, et al., Proceedings of the IEEE, Vol 89 No 3, March 2001，已获得许可

总体来看，处理器是所构建出的尽可能最有效地将微处理器系统从一个机器状态移动到下一个状态的状态机。机器的状态由其寄存器的容量决定：其移动到的机器状态由在寄存器间执行的指令指定。处理过程通过在取到较接近处理器处的系列

数据上执行的系列指令集实现。如果所需的数据不在本地存储，对其的调用被称为“未命中（miss）”。称为处理器工作负载的一串指令在本质上通常是科学型或者经济型的。这两种不同类型的工作负载在利用处理器资源时大相径庭。类似3D结构的启用在通用处理器方面是有用的，条件是通用处理器允许一个给定的设计能够很好地收回两种类型的操作。

由于性能组分的重要性，我们应检查微处理器吞吐量延时的产生者们。图1-6a示出了微处理器的基本元件。图中的指令单元（“I-Unit”）译码并发送指令到处理器，执行单元（“E-Unit”）执行这些指令，同时一级缓存阵列存储操作数[3]。如果执行单元没有延迟保护操作数，所需回溯指令的周期数将在图1-6b最低线处被捕捉，并标记E-busy（执行忙）。指令单元的数据仍需从一级缓存中再次取出。一级缓存此时极有希望被预期所需数据填满。在理想无限缓存的情况下，缓存无限大并且包含每个可能被请求的字。此时的性能将由第2条蓝色线确定。蓝色线包含了微处理器延时和到一级缓存的访问时间。然而一级缓存毕竟是有限的，同时被请求的数据也过于经常地没被正确预测并预装载到一级缓存之中。每条指令所需的指令周期在“有限缓存”的现实下包含了所需获取数据的未命中时间损失。

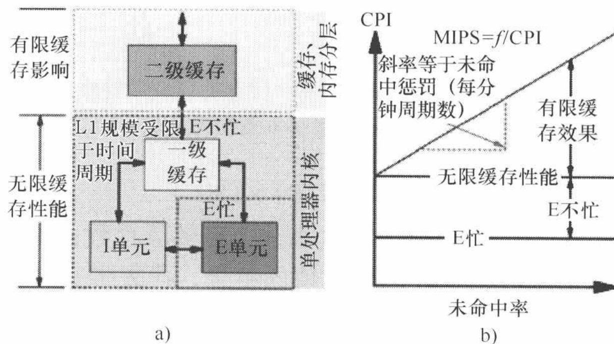


图 1-6　处理器性能组成。延时的决定因素依次为：a）理想处理器；b）到本地缓存的访问和缓存的重填充© 2006 IEEE 来自本章参考文献［3］，已获得许可

不论科学型或经济型，微处理器的性能同时依仗其有效数据输送能力和高性能逻辑。尽可能快地向存储器提供与延迟、带宽和缓存阵列相关的数据。数据请求到其有效的延时称为延迟。片上的临时缓冲存储器减少了部分片下主存储器数据传输需求。带宽允许在任意给定时间内取更多的数据到片上。最重要的，这3个处理器存储器子系统的属性是可交换的。

当片上系统缓存不能在片上继续增加时，改善带宽和到主存储器延迟的技术就变得非常重要。图1-7示出了假定的情况，其中在给定处理器上运行的线程或独立计算进程的数量翻番。为保持未命中率恒定，可观察到提供给芯片的数据量必须增加[3]，否则线程数量不可能增加。因为将出现更多未命中并使任何潜在优势失去

意义。如图 1-7 底部左侧所示，为新线程添加第 2 个缓存需要带宽翻番。换句话说，如果带宽没有翻番，那么每个缓存必须翻 4 倍来弥补每线程的网络带宽损失，如图 1-7 底部右侧所示。假定未命中率与片上缓存面积的二次方根成正比增加，则带宽和存储器面积的可交换度可归纳为

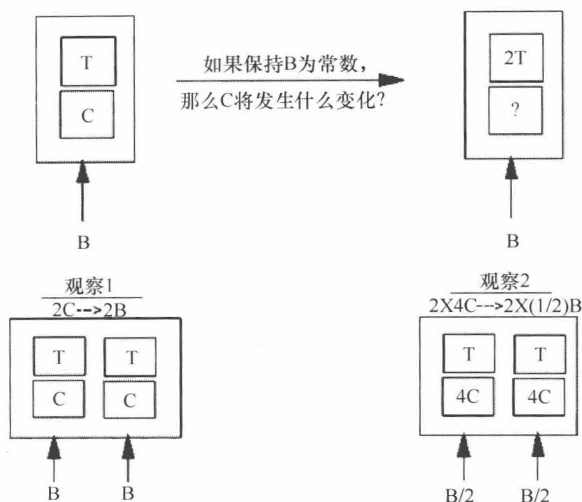


图 1-7　线程数翻番时，如果总带宽保持不变，缓存的容量需要翻两番才能保持每线程与之前相近的性能© 2006 IEEE. 来自本章参考文献［3］，已获得许可

$$2^{a+b}T=(2^aB)\times(8^bC) \tag{1-1}$$

式中，T 代表线程数；B 代表带宽；C 代表片上有效缓存数；指数 a 和 b 为假定的等价（此处为固定的未命中率）的任意值的组合，代表带宽和存储器面积的可替换关系。假定昂贵的未命中率与右侧第二项的存储器相关，并随带宽变化，像 3D 集成提供的一样，其在多线程的未来处理器中是有潜在优势的。

现在我们考虑在多种不同类型的工作负载下未命中的影响。科学型工作负载就像那些大型科研装置一样，是高度固定的操作模式在大量数据集上执行。处理器本地缓存阵列需要从主存储器获取的数据是非常容易预测和连续的，并且允许存储器子系统以最少的说明、中断和错误的方式将数据从存储器缓存送到处理器。未命中极少发生。在这种系统中，性能直接与将数据导入处理器的总线带宽相关。该总线本身具有高占用率并时刻保持满状态。总线不满时，系统吞吐量实际上降低了。另一方面，经济性负载具有不可预测性和不固定的数据模式。此类系统通常用来处理多种多样的处理任务。数据未命中经常发生，它们的发生概率服从泊松分布。图 1-8示出了以每两次未命中之间回溯指令数为自变量，总未命中数比例为因变量的曲线图。虽然图中的峰值是在 X 轴右端终点，但实际上未命中经常发生，且这是常识。高吞吐量需要低总线占用率以实现避免总线在未命中爆发时“卡住”。

图1-9示出了这种关系并展示了可访问低占用率总线对经济型处理器来说多么重要。当总线占用率超过30%，相关性能直线下降。

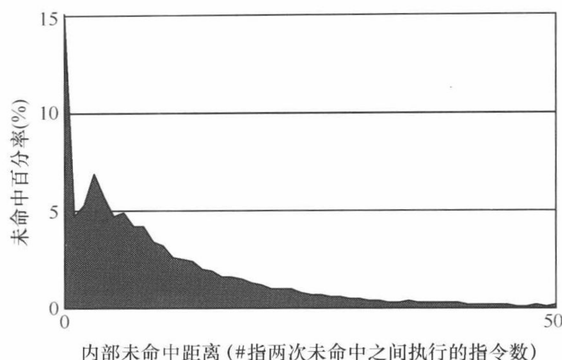


图1-8　以每两次未命中间回溯指令数目为自变量，总未命中数百分比为因变量的柱状图 © 2006 IEEE. 来自本章参考文献［3］，已获得许可

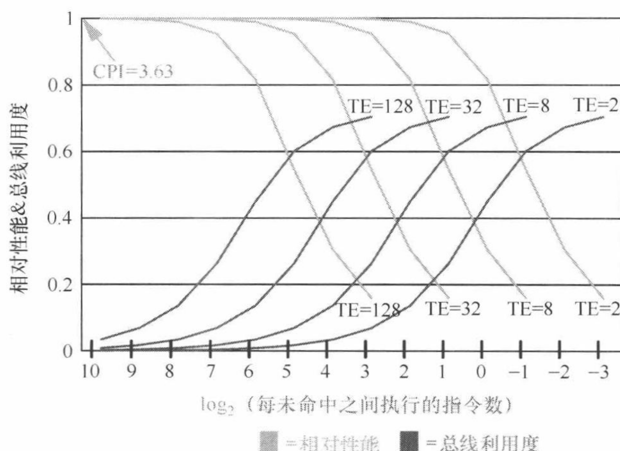


图1-9　相关性能和总线利用度© 2006 IEEE. 来自本章参考文献［3］，已获得许可

简而言之，两类应用空间都需要带宽，但原因大相径庭。假定处理器经常用在通用机器中，而该机器同时用于科学型和经济型的背景设置下，那么把两类工作都做好很重要。当一类技术解决方案强调其中一种时，3D集成由于其带宽优势成为一种常见的解决方案。

如图1-6所示，延时有很多原因，但它们全部不同程度地受互连延时的影响。在“无限缓存场景”下的性能由处理器自身的执行延时决定。当然，处理器延时本身随互连延迟的减小而改进，但在3D成功成为存储器到处理器的一种传输方式时，处理器执行延时本身改进很小。在有限缓存场景下，在我们必须将有限缓存和

带宽衍生的数据延时考虑在内，而这正是表明3D集成正处有用之时。在图1-10中，我们看到当单核、双核和四核处理器在3D集成工艺实现时获得的改善。系统中与结构相关的性能一直增长直到系统被获得足够数据供应的点，此时未命中率得到控制。超出这一点，再提供更多带宽就没有意义了。读者们应该注意，该饱和点随处理器核的增加移动得越来越远。意义在于，由于数据传输是未来多核处理器的重要属性，设计者仍将保证处理器核的性能持续改善。

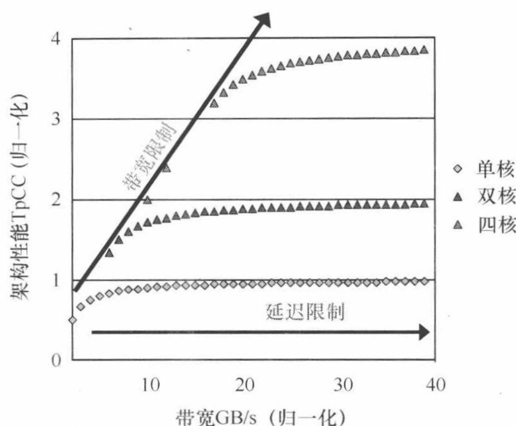


图1-10 不同处理器核的带宽和延迟边界

3D集成的中心问题是需要面对一种最终总线的概念。“总线占用”是一个问题。该问题的症结如下：得到一条处理器所需数据的时间决定了系统数据延迟。像我们已经了解到的，延迟直接影响了等待该数据的处理器的性能。另一个，总线相关的不易察觉的性能影响仍然在总线被数据传输占用时出现。如果运气好，得到的整条数据线是所需的和有效的。然而常常只有这条线的第一部分是有效的。由于数据必须通过“单线地址累加”进入处理器，意味着最少读入一整条数据线。微处理器架构师困扰于一次应该从主存储器中获取多长的数据线。如果太短，所需的总线容量可以更精确的确定，但微处理器本地缓存中所含数据的片上地址目录空间将非常大。而太长的数据线削减了系统中预订—保持和存储器管理成本，但在并发未命中需要获取新地址时，应使总线忙于下载更长的数据线。这中间是矛盾的。数据线的长度因此根据微处理器被调用执行特定应用程序时工作负载的分布统计确定。图1-11示出了该动态关系。当一个事件引发缓存未命中时，存储器子系统迅速向缓存传回数据的过程是存在延时的[3]。最终，带着由数据线首边沿到达定义延迟的数据开始抵达。然而直到尾边沿抵达前，即总线在整条数据线进入完成前不再空闲。总线被占用的时间越长，下一个未命中被处理需等待的时间越长。3D的价值在于更大的带宽（通过更宽的总线）减轻了由于尾边沿影响引发的延迟。

接着，需要指出工业趋势显现出每代产品中处理器核的数量以二次方率增长。

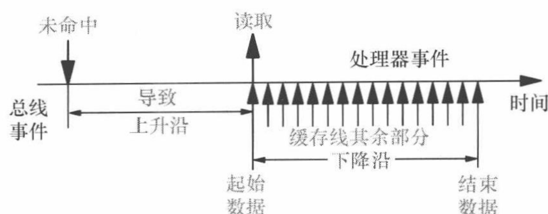


图 1-11 缓存未命中占用总线直到所需的整个缓存线传输完毕，同时阻塞了其后的数据请求 © 2006 IEEE. 来自本章参考文献［3］，已获得许可

图 1-12 示出了微处理器时钟频率（其驱动数据需求）、存储器时钟频率和存储器总线带宽增长率的差异。数据总线频率传统上随 MPU 频率以 1:2 的比率变化，每 18～24 个月翻番。而数据总线带宽以低得多的速率（注意纵轴是对数坐标）增长，说明数据总线传输率的主要改善源于频率提高。当且仅当时钟频率变慢时，除非我们替换总线的带宽技术，不然数据总线容量必然超标。可以说带宽的杠杆作用是设计者在提高处理任务率时所强调的结构范例序列中的最后一个。早些时候的手段倾向于强调片上的其他资源，例如像图 1-13 所示的，增加所需的寄存器数目。现如今，轮到了带宽。再次集成到 Z 平面推迟了互连相关的限制从而延长了经典的按比例缩小。但与此同时，有些情况发生了实质改变。本书其他部分的探索给出了向另一个维度扩展的解决方案。对于认知生物学函数而言，该方案是自然已展示出其必要性的解决方案。

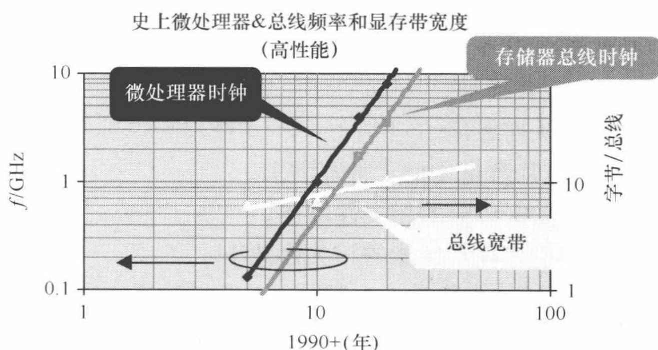


图 1-12 频率驱动的数据速率：数据总线频率随 MPU 频率以 1:2 的比率变化，每 18～24 个月翻一番，而数据总线带宽只是较平缓地增长。数据总线传输速率基本上通过总线频率按比例缩小：当时钟增长减缓时，总线数据速率增长同时放缓

总结所述结构问题，以下几点应牢记：

- 频率不再增长。
  - 逻辑比存储器总线更快地按比例缩小。
  - 处理器时钟和总线时钟消耗带宽。

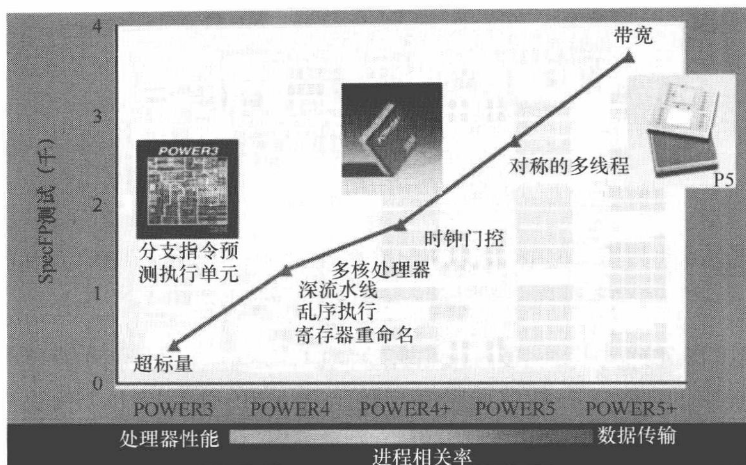


图 1-13 功耗系列：结构性能分布图

- 更多的指令预测使预取尝试次数加倍增加。
  - 错误的猜测增加了未命中容量。
- 减小数据线长度在缓存增长时受到目录限制。
  - 因此，加倍数据线长度使总线容量翻番。
- 每块芯片上处理器核数目 N 随每代更替而增加。
  - 使片下总线任务以 $N/2 \times \mathrm{Sqrt}(2)$ 倍增长。
- 由此导致每处理器核的更多的线程和虚拟化的增加。
  - 使片下总线任务以 N 倍增长。
- 处理器/SMP 总数在增加。
  - 使整个系统的队列恶化。
- 增加处理器核/芯片数量增加了带宽需求。
  - 任务回溯率与数据传输的相关性增加。
  - 任务回溯率与单处理器性能的相关性减小。

以上的讨论把 3D 集成技术作为一个整体。实际上，我们以上述展示出的结构优势将随着 3D 技术的进步而逐步实现。“三维”实际上指一系列工艺和能力，其及时发展出更小的通孔间距，更高的通孔密度和更低的通孔阻抗。图 1-14 给出了一旦达到某能力阈值，可应用 3D 技术的分布示意图。

在本书随后的部分中，几位 3D 集成专家将教我们怎样不仅在前面讨论的存储器子系统中，而且在任意新的、创新的应用程序中利用该技术。未来工作的挑战是巨大的，工艺和技术制程固然需要实现，但所需的潜在基础也不可忽视：EDA、测试、可靠性、封装，以及其他我们在 2D 集成电路认为理所当然可以工作的配备技术。但在执行好的情况下，得到的计算密度和它们支持的新能力将是惊人的。即使

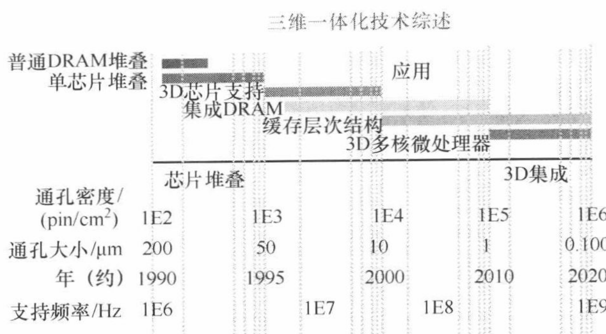


图 1-14 3D 集成技术图谱

仅仅之前在本章讨论的存储器管理的例子中，3D 技术保证了的大量存储的实时访问将被视为我们工业的分水岭。

本书的后续部分从第 2 章 3D 工艺的简要介绍开始，其作为设计者理解 3D 制造方法（对 3D 工艺全面的细节，可以参看本章文献［4］）的简要参考。本书的下一部分将重点关注 3D 集成电路设计的自动化工具，包括热分析和电源传输（第 3 章），基于热分析的 3D 布局规划（第 4 章），基于热分析的 3D 布局（第 5 章），以及基于热分析的 3D 布线（第 6 章）。随后是 3D EDA 工具的讨论，后三章给出了 3D 微处理器设计（第 7 章），3D 片上网络芯片结构（第 8 章），以及用于高能效服务器设计的 3D 堆叠应用（第 9 章）。最后本书以关于 3D 集成电路技术成本影响的第 10 章为结束。

参考文献

1. S. Amarasinghe, Challenges for Computer Architects: Breaking the Abstraction Barrier, *NSF Future of Computer Architecture Research Panel*, San Diego, CA, June 2003.
2. J. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. Saraswat, A. Rahman, R. Reif, and J. D. Meindl, Interconnect Limits on Gigascale Integration (GSI) in the 21st Century, *Proceedings of the IEEE*, 89(3): 305–324, March 2001.
3. P. Emma, The End of Scaling? Revolutions in Technology and Microarchitecture as We Pass the 90 Nanometer Node, *Proceedings of the 33rd International Symposium on Computer Architecture*, IBM T. J. Watson Research Center, pp. 128–128, June 2006.
4. P. Garrou, C. Bower, and P. Ramm, Handbook of 3D Integration, Wiley-VCH, 2008.
5. D. Matzke, Will Physical Scalability Sabotage Performance Gains? *IEEE Computer*, 30(9): 37–39, September 1977.
6. F. Mujica, *History of the Skyscraper*, Da Capo Press, New York, NY, 1977.
7. V. Srinivasan, D. Brooks, M. Gschwind, P. Bose, V. Zyuban, P. Strenski, and P. Emma, "Optimizing Pipelines for Performance and Power," *Proceedings of the 35th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 333–344, November 2002.

第 2 章　3D 集成电路工艺考量

Albert M. Young Steven J. Koester

集成电路产品尺寸小型化和功能全面性的发展趋势有力地推动了 3D 集成电路技术的快速发展，并使其具有广阔的市场前景。虽然 3D 集成电路工艺还没有在整个产业中实现标准化，但对于 3D 集成电路的设计者，及时了解 3D 集成电路的工艺发展趋势及其带来的盈亏和利弊，至关重要。在本章，我们涉及设计者关心的基本工艺问题，包括各层的堆叠方向，各层的对齐，粘合接触面的设计，硅通孔的尺寸，以及与 CMOS 工艺的集成。以上这些都是与设计直接相关的问题，对设计者在实际工作中选择何种 3D 集成电路工艺，以及在给定的 3D 集成电路工艺下电路的优化设计非常重要。

2.1　介绍

集成电路产品尺寸小型化和功能全面性的发展趋势有力地推动了 3D 集成电路技术的快速发展，并使其具有广阔的市场前景。虽然 3D 集成电路工艺还没有在整个产业中实现标准化，但几乎所有 3D 集成电路工艺都具有以下关键技术：垂直硅通孔互连、粘合对齐、基于背面工艺的晶圆减薄。我们希望本章可以使设计者更好地了解 3D 集成电路工艺的发展趋势，因为此发展趋势对 3D 集成技术的进步和电路设计本身均有影响。

在过去的几十年中，我们看到了计算机性能的巨大飞跃。这主要是半导体器件特征尺寸不断减小的结果。从 20 世纪 60 年代到今天，半导体器件的特征尺寸已经从毫米级缩小到纳米级，从而使芯片上的晶体管数目成几何倍数增加，几乎每 18 个月翻一番。Gordon Moore 最早预测了这种增加趋势，并由此命名为摩尔定律。我们不能低估这种趋势，因为它所带来的计算机性能的提高极大地影响了社会发展的方方面面。

随着摩尔定律的成功实现，硅基 MOSFET 的特征尺寸持续减小，未来集成电路的集成度将不断增加[2]。但是，当今集成电路工艺的发展遇到了几大瓶颈。第一，小尺寸器件的光刻花费巨大，这意味着不断减小的光刻尺寸将产生高额的成本；第二，如果光刻尺寸能够不断减小，集成电路的时钟频率应等比例增加，但芯片上高密度的晶体管所产生的大量功耗会极大地抑制时钟频率。事实上，当今处理器设计主要依靠高效新颖的架构提高性能，而非简单的时钟频率。这就意味着，将来的集成电路的发展只单独依靠减小集成电路的特征尺寸和增加集成度，很难再提

高系统性能，人们需要新方法来实现设计目标。因此，3D 集成电路技术应运而生，它不依赖于器件尺寸的减小，其特点是：①通过 3D 集成减小总的互连线长度，从而有效减小互连延迟；②各层之间的互连急剧增加；③不同材料、工艺和功能的模块集成于一体。

总的来说，3D 集成技术可以广义地定义为：基于任何工艺的半导体元器件层层向上堆叠，并利用垂直的，而非外围的互连线，彼此相连。根据这一定义，3D 集成技术内容广泛，包括简单的芯片堆叠，硅基芯片载体与嵌入体的堆叠，芯片与晶圆的堆叠，以及完整的晶圆级堆叠。每种堆叠针对特定的应用，呈现出不同的优点。而互连线密度决定了在某种特定应用下，采用何种堆叠技术。例如，在无线通信中，芯片只需少量硅通孔来连接地线，从而降低电感；在高性能服务器和多层堆叠的存储器中则需要极高密度（$10^5 \sim 10^6$pins/cm^2）的垂直互连线；而在 3D 芯片中电源稳压器和调节器设计在中心位置，满足了全方位的高密度互连的需要。

3D 集成工艺技术方法多样，并已有相关文献给予过详述[3]。这些工艺方法有着各自的优点和利弊，各种不同的应用需求最终决定了采用何种工艺流程。然而，几乎所有的 3D 集成工艺都有 3 个主要的工艺流程：①垂直互连；②粘合对齐；③晶圆减薄与背面处理。这些流程的顺序在不同的工艺中各不相同，它的选择在很大程度上取决于最终应用。而工艺选择也会最终影响整体设计，我们将在后面的章节予以讨论。为了更好地明白 3D 集成电路的先进性，首先我们应了解今天 3D 集成技术一些初期的产品应用。

2.2　背景：3D 集成技术的初期需求

移动便携设备小型化的趋势是推动 3D 集成技术发展的初期市场驱动力。作为一项关键的产品——CMOS 图像传感器（摄像机，蜂窝网通信手机中使用的模块），带动了晶圆级封装（WL-CSP）技术的发展。Shellcase 公司（后来又收购了 Tessera 公司）在晶圆级封装技术方面实力雄厚。他们的很多技术可以对比 3D 集成技术：①晶圆级封装技术不是电路级的堆叠，而是晶圆级；②晶圆级封装技术经常在芯片边缘的部分布线，从而完成前后晶圆的电学互连。然而，WL-CSP 并没有带来期望的技术进步，没有推动诸如硅与玻璃的粘合及随后的晶圆减薄等技术，而这些技术恰恰是 3D 集成技术的主流技术，沿用至今。在另一个领域，多芯片封装（MCP）技术利用多层芯片实现了存储器的海量集成，发展迅猛。而且，这项技术推动了高可靠性的晶圆减薄技术，推动了不依赖于隔离层的芯片堆叠技术。以上这些封装技术，实现了层与层之间的引线互连，都为移动便携设备小型化提供了卓有成效的解决方案。但是，由于移动便携设备的不断发展和进步，新的 3D 集成技术呼之欲出。

便携设备上越来越多的附加功能，使人们对设备的要求不再局限于简单的外形

尺寸的小型化，而是性能的全面提高。例如，高端的应用程序所要求的运算速度远远超出传统引线带宽可支持的范围。另外，对信号总带宽不断地增大则要求互连线由原来的边缘部分延伸至（如芯片边缘或引线键合互连）整个芯片。这就需要能够贯通整个堆叠芯片的新型技术，即硅通孔和平面阵列互连。以前，由于采用硅通孔的产品成本较高，硅通孔的应用步伐逐渐放缓，而且，硅通孔应用也主要针对I/O较少的情况，放置在电路外围边缘，使其不能发挥固有的优势。但在当今设备高性能的要求下，硅通孔从芯片边缘逐渐过渡到整个芯片。而在微机电系统（MEMS）制造中深度反应离子刻蚀工艺技术的问世，使硅通孔可以借用此技术占用较少的芯片面积，减小了使用硅通孔的成本。而芯片与芯片之间的平面阵列互连，已应用在索尼的游戏掌机（PlayStation Portable，PSP）中。在可接受的成本下，提高了内存和处理器之间的通信带宽。他们采用了一种基于微凸焊的技术，提供高带宽连接（30μm焊球和60μm间距）和两芯片有源层面对面互连，实现倒装封装。这种方式虽然实现了两层芯片的高带宽互连，但是，如果没有硅通孔，这项技术便不能扩展到两层以上多层芯片间的通信。因此，结合具有精密间距硅通孔的平面阵列互连，显然是下一代3D集成电路技术的主要发展方向。

晶圆与玻璃键合，晶圆减薄和芯片边缘插入硅通孔等技术业已应用在当今大规模的产品制造中。但是，下一代3D集成电路技术会朝着不同方向继续发展，这使制造工艺对产品的影响更加复杂，需要认真研究。下一节，我们将确定需要考虑的工艺参数。

2.3　影响3D设计艺术状态的工艺因素

由于现今已有的许多3D集成电路制造技术都有各自的工艺参数，并在诸多方面影响了3D集成电路的设计，所以了解设计和工艺之间的关系非常重要。在这里，我们就一些关键工艺参数如何影响设计，提供指导性意见。这些意见包括各层的堆叠方向、对齐的规则、键合界面的设计，以及硅通孔的设计要点和工艺整合。

2.3.1　各层的堆叠方向：正面对背面与正面对正面

在3D集成电路的堆叠结构中，各层芯片的堆叠方向对设计至关重要。它们会影响不同层晶体管的间距，并涉及EDA工具中的设计镜像问题。当多层芯片堆叠在一起，任意两层的堆叠组合有多种，但是正面对背面和正面对正面的比较可以说明许多重要的问题。这两种选择的图示说明如图2-1所示。

2.3.1.1　正面对背面

正面对背面的方法是使上层芯片（经过减薄）的底部和下层芯片的顶部键合。类似的方法最初是应用在IBM公司开发的G5系统中的多芯片模块（MCM）[4]，后来同样的方法在CMOS和MEMS的晶圆级设计中同样适用。图2-1a示出了两层堆叠芯片正面对背面的结构。因此，该结构的厚度依赖于减薄后上层芯片的厚度。如

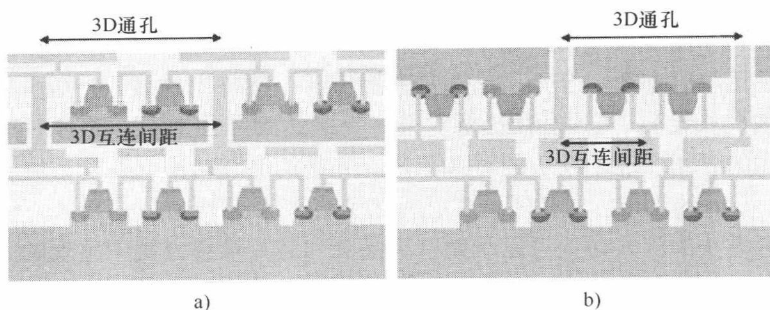


图 2-1　两层芯片 3D 堆叠结构的示意图：a）正面对背面；b）正面对正面。3D 硅通孔的间距与 3D 互连线的对照

果由于工艺的限制硅通孔的长度必须和截面积呈比例，那么衬底的厚度就直接影响两层晶圆间互连的数量。同样重要的是要注意，在这个结构中，晶圆之间的互连的总数不能大于硅通孔的数量。为了实现这样一个堆叠，需要利用承载衬底层，众所周知，承载衬底可能使晶圆顶部发生扭曲，难以实现较小的对准误差。在以前的工作中，我们发现这种由承载衬底层引起的扭曲达到 50×10^{-6}，但通过诸如具有温度补偿的键合可用于减少这些扭曲。目前的正面对背面结构中，典型的上层芯片的厚度为 25 ~ 50μm，这使通孔和互连间距限制在 10 ~ 20μm，但将来晶圆减薄和通孔填充技术的进步会减小这一数值。

2.3.1.2　正面对正面

图 2-1b 示出了正面对正面的方法，即把晶圆的器件层键合在一起。这种方法最初由 IBM 公司提出[5]，在 20μm 的互连线间距下，设计多芯片模块，与正面对背面结构相比，减小了工艺的复杂性。正面对正面结构主要的潜在优势是使层和层之间硅通孔的数量成倍增加。因此，与正面对背面结构相比，它可以实现更高的互连密度。在这种情况下，互连间距仅由键合工艺的对准误差限制（加上标准 CMOS 光刻步骤引起的正常套刻误差）。在当今最先进的对准键合系统中，典型的误差为 1 ~ 2μm，因此正面对正面结构能够实现 10μm 间距，甚至更小。然而，这种层间连接密度的提高，只能局限在两层堆叠结构中；对于多层堆叠结构，硅通孔仍然会限制总的 3D 互连密度。

2.3.2　层间对准：层间互连误差

从上面的讨论可以看出，层间对准误差会直接影响 3D 集成电路的层间互连线密度，并进一步影响 3D 集成电路的整体性能。误差由工艺决定，不同工艺的误差区别很大，所以工艺加工能力对误差大小至关重要。例如，不同层芯片的对准误差范围从 1 ~ 20μm 不等，这主要取决于工艺加工的速度。在制造工艺中，可以精确地把握层间互连的间距，但是对准每层芯片所需的大量时间会给制造工艺带来极大

的挑战。在未来几年，对准工艺带来的挑战会逐步解决，从而使以上方法可行。未来，尽管层间互连线尺寸不断减小，但层间距可以精确的把握，这为高性能芯片堆叠提供了最佳解决方案。今天，晶圆对准提供了一种替代方案，晶圆由很多芯片组成（芯片彼此相连），可以利用晶圆对准，一次达到晶圆上所有芯片对准。这就允许花费更多的时间专注于精确的对准工艺。先进的晶圆对准和键合方法目前通常可以实现 1 ~2μm 的误差。虽然不同工艺所带来的一系列问题是复杂的，但下面对晶圆对准键合方法的深入讨论可以帮助解决晶圆和芯片堆叠方法中出现的问题。

在 3D 集成电路中，对准晶圆键合与传统的键合技术不同。例如，与 SOI 工艺中衬底的制造相比，有很大不同。第一，为了达到预期的层间互连密度，发挥 3D 集成电路的特有优势，需要记录每个晶圆刻蚀的图案来实现对齐。第二，3D 集成的每层晶圆表面通常有显著的凹凸，它们可以用来完成高品质的键合，但与平整键合相比，特别是对于氧化物融合键合，显著增加了难度。最后，因为 CMOS 电路（通常后段工艺金属化）已经存在于晶圆上，在键合过程中的温度限制非常严格，典型的键合温度通常低于 400℃。

在 CMOS 制造中，晶圆对准从根本上不同于今天通常使用的基于步进重复投影曝光的光刻对准。这是因为晶圆对准必须在整个晶圆进行，而不是基于芯片之间。和芯片级对准方案相比，覆盖控制更加困难。非理想因素，如翘曲度、光刻歪斜或跳动、热膨胀都能导致覆盖误差。此外，基板的透明或不透明也可以影响晶圆对准。设备制造商已经为 200mm 和 300mm 晶圆开发了对准精度在 1 ~2μm 范围内的对齐工具。

由于温度波动和与键合工艺本身相关的潜在扭曲存在，在工业界的标准程序中，首先应使用对准设备（具有高吞吐量），然后将晶圆移动到特殊化的键合设备，来更好地控制晶圆和 3D 堆叠间的温度和压力。良好的工艺控制的关键是将对准和键合前的步骤从实际的键合工艺中分离开来的能力。这种集成一体化的设计可以更好地了解最终的对准误差。也就是说，实际使用的键合工艺和技术能影响整体的对准覆盖。理解这个问题对于正确选择键合工艺是很重要的。

例如，当来自两个衬底周围的绝缘材料凹进时会产生一个铜-铜键合对准问题。在这种情况下，如果有键合前的大偏差，晶圆仍然可以连接压焊线。此外，这种结构无法抑制由于热压缩而产生的热偏差，不能抑制由于热压缩过程诱导的剪力产生的对准滑移。为了防止这种滑移，一种方法是使用锁-钥匙的结构在表面限制偏差量，即在初始对准和布局步骤中，保持一个晶圆对准到另一个。这是在保持 3D 工艺更紧密互连间距能力的一个重要因素，因为可允许间距往往受限于对准和键合公差。

在减薄工艺中，晶圆处理和分层工艺会给减薄的硅层增加扭曲。这种扭曲由材料间的热膨胀系数的不同和低弹性的聚合物复合材料的使用引起。作为一个例子，如果不加以控制，玻璃化晶圆的使用会在 200mm 的晶圆边缘引起 5μm 的对准误差，这明显大于直接的硅-硅对准误差。所以，对于工艺中使用的处理后的晶圆，控制和减小这些误差是重要的考虑因素。在实践中，这些扭曲往往可以作为全局误

差来建模，从而我们就可能基于温度控制、晶圆材料和层压聚合物来纠正大部分的晶圆级误差。

对准有时候仅仅用于基于 SOI 的氧化物键合方面。这个过程经常用在 SOI 晶圆被夹在玻璃化晶圆中，并且硅衬底除去之后，留下一层 SOI 薄层附着在玻璃上。不像其他的两种情况，一是单独的光路用做对准晶圆表面成像况，二是需要用红外成像通过晶圆堆叠层，对于 SOI 晶圆对齐，人们可以通过可见光波段看到这种类型的样品。这允许非常准确的直接光学对准到下层的硅晶片，用一种和晶圆级片规模对准器的方式。晶圆片接触和初步氧化物熔接必须由对准工具本身启动，然而一旦实现这一点，会在下游公艺中产生小的最小对准失真。

2.3.3 键合界面设计

叠层之间高质量的键合界面设计需机械的、电气的和热的仔细分析。在下一节中，我们简要地描述 IBM 公司研究的 3D 晶圆键合对准的 3 个特别的技术：①铜-铜压接；②传递和交接键合（混合铜和粘黏键合）；③氧化物熔接。类似的对于芯片级叠层间的设计需要关注。这些硅芯片之间往往需要焊料和填充材料。

2.3.3.1 铜-铜压接

通过高温表面铜化的两个晶圆间的压力，使两个晶圆的压接成为可能。对于 3D 集成，铜—铜连接可以为两层之间提供额外的电气连接的功能。该键合过程的质量优化是正在解决的一个关键问题，包括多种平面制备技术、键合后矫直、热退火循环，以及使用优化的图案几何技术[8-10]。

铜热压接发生在高温度和高压力的情况下，此时铜区域之间的微观接触开始变形，进一步增加了其接触面积，最终扩散到彼此以完成键合工艺。铜键合的关键参数包括键合压力、温度、持续时间，以及铜表面清洁度。为了获得高质量的键合，优化这些参数是必需的。表面清洁程度不仅和键合前表面清洁有关还和键合过程中的真空条件有关[8]。另外，尽管事实上键合温度是决定键合质量的重要参数，但温度必须与后线工艺温度相容以不影响器件的性能。

对于真实器件的晶圆级铜键合的质量，在本章参考文献［9，10］中有所研究。铜键合模式的设计不仅影响电路布局，而且影响键合质量，因为它和可用面积相关，此可用面积的区域是在一个本地的区域或穿过整个晶圆而被键合的区域。铜键合衬垫的大小（互连的大小）、衬垫密度（总的键合面积）和密封设计也已经被研究。基于不同的铜键合密度的研究表明，更高的键合密度会有更好的键合质量，并且可以在切割测试中出现很少的失效。另外，一个在电互连周围、芯片的边缘和晶圆边缘具有额外的铜键合面积的密封设计能够防止腐蚀和提供额外的机械支持[9]。

2.3.3.2 混合铜/粘黏键合（传递-压接）

关于铜—铜压接工艺的变形可利用具有中间粘合层的一种锁-钥匙的结构实现来提高压接的强度。这项技术最初是为 MCM 超薄膜模块开发的并且在构建和验证

期间进行了大量的可靠性试验[11,12]。然而，正如之前提到过的，该方案同样适用于晶圆级 3D 集成，并可能相对于基于铜-铜直接键合的方案有显著的优势。

在传递-压接组件方案中，将连接在一起的两个晶圆的表面，一面设置有一组突起（键），它将匹配到另一面的插座（锁），如图 2-2a。一个突起，也被称为一个螺柱，可以是一个硅通孔的延伸或一个特殊制造的后线铜螺柱。插座设置在底部，具有一个铜垫片，铜垫片会和铜螺柱键合在一起。至少所述表面中的一个（图 2-2a 中下面的一个）在钝化电介质层顶上设置有粘合剂。两个衬底可以是硅衬底，或其中之一可以是减薄处理后的晶片。在每个晶圆，螺柱和衬垫被选择性地连接到电路，并通过适当的 2D 互连和/或硅通孔互连。

在铜-铜直接键合技术中，这些衬底可以通过一种均匀的理想的压力，并在 350~400℃的温度范围下对齐然后键合到一起，螺柱的高度和粘合层或绝缘层的厚度匹配，以使得在键合过程中，首先实现铜螺柱到铜焊盘的接触。在持续的键合压力条件下，螺柱的高度被压缩，粘合剂加入到接触面中，和相应的绝缘体表面键合。

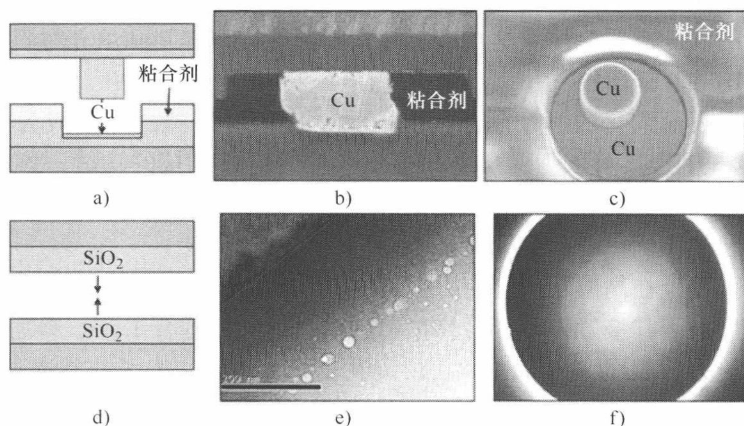


图 2-2 键合方案：a）传递-压接键合方案截面图；b）抛光的传递-压接键合方案截面图；c）自上而下的简化层次后的传递-压接键合扫描电子显微镜图像；d）氧化物融合键合方案；e）透射电子显微镜（TEM）下氧化物融合键合的横截面图；f）用于两片晶圆的氧化物键合的晶圆红外图像

粘合剂材料按满足合适的流变力学的需要而选择，通过填充在晶圆之间的间隙，从而使两个晶圆键合在一起。此外，该粘合剂在键合温度条件下和任何后续处理情况（附加层连接，3D 堆叠最后的后线布线等）所需要的条件均为热稳定的。无论是承载晶圆已进行处理或进行了背侧晶圆减薄，都会进行键合处理。如果考虑额外的晶圆层连接，该过程可以根据需要重复。

一个完整的具有聚合物胶粘剂层的铜-铜传递-压接键合如图 2-2b 所示。这是在一个键合结构中，在上层衬底被减薄后，为了构造上的分析，图 2-2c 另外展示了螺柱到垫片的对齐。这种锁-钥匙的转移交接的方法可以和任何之前描述的 3D

集成方案结合。粘合剂增加了 3D 堆叠的机械完整性，意味着铜—铜直接键合的铜密度的要求可以放宽，甚至舍弃。

2.3.3.3 氧化物融合键合

氧化物键合可以用来将两个全工艺的晶圆相连在一起。在 IBM 公司，已经公布了大量的关于这种基本工艺的使用方法。这种工艺可以将在面对背方向的 SOI 衬底连接在一起[13]。在 3D 集成电路中，使用氧化物融合的其他方案已经由他人实施[14]。为了保证集成电路键合平面平坦化特点和这些表面的活化性质的兼容性，通常的要求包括低温键合氧化物的沉积和退火，以提供适当的化学性质，允许粗糙的键合发生。氧化键合工艺的示意图如图 2-2d 所示，接着是键合界面的横截面透射电子显微镜照片（TEM）（见图 2-2e）和典型键合（见图 2-2f）的整个晶圆的红外图像。透射电子显微照片显示了分布的微空洞模式，而平面图红外图像表明，经过 150℃ 和 280℃ 键合后退火，优良的粘接质量得以维持，虽然偶尔观察到宏观尺度的空隙。

采用多层次的后端布线通常会导致显著的表面形貌。这为需要极其平坦表面的氧化键合带来了挑战。虽然在掩膜设计中，通过控制金属图形的密度可以减少非平坦化，但也需要基于工艺的平坦化方法。正如本章参考文献［15］所述，具有后端金属化的典型晶圆有显著的模式诱导的拓扑结构。我们已经展示了在一个高度优化的化学-机械抛光（CMP）后，先进的平坦化方案合并厚的 SiO_2 层的沉积，可以大大降低依靠图形模式造成的变化，这是需要达到的良好的键合效果。这种先进的平坦化技术的发展将是氧化键合方案的商业化的关键，但在常规基础上可能遇到图形模式有关的拓扑变化。而将这一类型的技术进入生产带来很多挑战，用氧化熔融键合连接 SOI 晶圆，它可以使设备之间有非常小的距离，同样可以导致非常高的密度互连。

2.3.4 硅通孔维度：设计点选择

也许对 3D 集成电路，最终要的拓扑元素是垂直互连，例如硅通孔。早期，硅通孔由像 IBM、东芝和意法半导体等公司引进到生产中，使用不同的金属材料包括钨和铜。对于 3D 集成电路，需要一个高性能的垂直互连才能真正地实现 3D 系统的性能优势。早期，由于互连被限制在芯片的外围，且比传统的平面技术难度更大，而不能显著地提高互连密度。在一定的芯片面积内得到硅通孔互连的方法和后线（生产线后端）半导体工艺有所相似。不同之处在于，对于每个硅通孔通常需要一个深的洞，这些通过硅衬底的洞不得不使用特别的刻蚀工艺被垂直地创造。

硅通孔的维度是 3D 电路设计的关键，由于它们直接影响了不可布线区域，这些区域是设计者不能放电阻的。在某些情况下，在工艺流水线后端也不能布线。然而，硅通孔的维度很大程度依赖于用于制造它们的 3D 工艺技术，并且更具体地说是硅层厚度、深高比，侧墙倾斜度和其他工艺特征参数的函数。这些维度也很大程度上受用于填充通孔的金属化的影响。这里，当处理两种重要的可相互替代的金属化材料时，我们将看到这些工艺参数选择的影响。

2.3.4.1 基于钨和铜硅通孔的设计考虑

1. 晶圆减薄的影响

晶圆减薄是3D集成电路的重要组成部分，因为它使得层间的距离减小，所以允许高密度的垂直互连。晶圆减薄的最大挑战是衬底必须减薄到原厚度的5%～10%，且需要小于1～2μm的一致性要求。在体硅中，由于刻蚀不会自然停止，这种减薄是特别具有挑战性的。最终的厚度取决于减薄工艺控制能力和硅移除工艺（机械网格和抛光，湿发或干法刻蚀）对厚度的限制。对一个几微米均匀厚度的硅片减薄已经成功验证，但是大于20μm的典型厚度需要更加先进的工艺。硅减薄的标准工艺步骤如下：第一步粗研磨，以便进行薄晶圆，从其原始厚度（700～800μm）减薄到125～150μm。这种工艺通常使用一个400目研磨表面进行。下一步用1800～1200目表面精细研磨，减薄到小于100μm。最后是机械抛光步骤，能减减薄到30～60μm的理想厚度。对于大多数过程，由于压力和机械研磨抛光会损伤硅的特性，这些只在硅的均匀区域进行的研磨步骤是理想的。如果有必要从背面露出硅通孔，通常希望使用一个基于等离子体的蚀刻工艺（如反应离子蚀刻）以暴露硅通孔，来完成减薄工艺。背面减薄的均匀化，限制了硅通孔深度，因此，通孔刻蚀可以使用此减薄技术。在通孔从衬底背面暴露后，将使用与晶圆级封装的工艺相似的技术来制造重分布层。某些情况下，可以制造多层布线。

不同的晶圆厚度可以导致不同几何形状的通孔，如图2-3所示。非常薄的晶圆，硅厚度小于30μm，由钨或铜制成的金属化单导体制成的全填充通孔可以以一个相当简单的方式实现。然而，在晶圆厚度更大的情况下，这里指大于100μm的情况，通孔的形式会不同。对于铜通孔，硅的厚度对硅通孔在晶圆表面占的面积有十分直接的影响。更厚的硅衬底导致更大的硅通孔所占面积，这将在某种意义上限制了成本，并且基于铜的通孔具有小的深宽比。随着晶圆厚度的增加，为了保持可制造的铜垫底的厚度，有从全电镀到部分电镀的趋势。钨通孔所占的面积对硅层的厚度并不敏感。虽然钨填充通孔的维度可以被薄膜淀积的厚度限制，用钨通孔的深宽比是很大的。一个宽范围的硅层的厚度通常可能有小的硅通孔占用面积；IBM公司经常制造超过100μm的钨硅通孔。

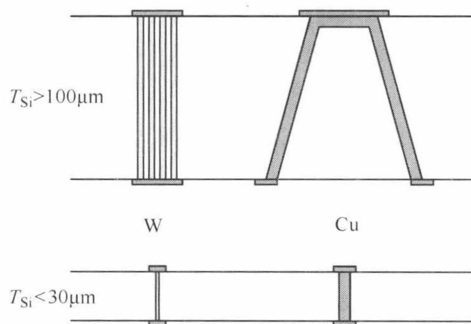


图2-3 不同的硅的厚度对钨和铜在不同情况下通孔的几何形状的影响

2. 对通孔电阻和电容的影响

用在硅通孔工艺上的传导金属的选择直接影响通孔的重要参数（如电阻和电容）。不仅不同的金属有不同的电阻率，而且它们各自的工艺局限也和它们可能的

几何形状相关。由于金属的选择直接影响通孔的比例（如深宽比），它还直接影响通孔的电阻。钨通孔通常沉积成的薄膜具有非常高的深宽比（远大于 20:1），因此很窄，往往具有相对高的电阻。为了减小这种影响，多个钨导体可以平行地绑在一起，以提供适当低的总电阻的层间连接，但以增长的面积为代价，如图 2-3 左上方所示。铜比钨有更好的电阻值。镀铜通孔的几何形状有低的深宽比（通常从 6:1 到 10:1），因此它们能够有很好的低电阻连接。

通孔电容很大程度受侧墙倾斜角度的影响。通常，钨通孔有将近垂直的侧墙并且没有显著的倾斜；铜通孔可以更容易地从倾斜侧壁获益。虽然侧墙倾斜增加了额外的通孔所占晶圆表面的面积，但锥形侧墙的引入可以帮助提升镀铜的质量，增加通孔隔离绝缘层的淀积率。这些淀积率受通孔的几何形状和淀积方法的强烈影响。这些方法能帮助增加最终通孔隔离层厚度，使得通孔有更低的电容，从而提高层间通信性能。

2.3.4.2　基于 SOI 3D 集成中极高的通孔密度

利用一个埋层氧化物 SOI 晶圆作为一种加强 3D 集成工艺的方式是可能的，如图 2-4 所示。这种所谓的 SOI 方案在之前发表的文章中已经被广泛地描述了[13,16]，这里仅进行简单的总结。不像其他传统的 3D 工艺，在基于 SOI 晶圆的 3D 集成方案中，埋层氧化物可以作为晶圆减薄工艺的最终停止层。这使得衬底在两个晶圆结合之前被完全移除。纯粹的湿化学蚀刻工艺均可使用。例如，TMAH（四甲基氢氧）可以有 0.5μm/min 的硅去除速率，且对氧化硅具有优良的选择性。在工艺中，通常通过机械技术移除 600μm 的硅晶圆，然后在 80℃ 的情况下使用 25% 的 TMAH（40μm/h 的刻蚀速率）刻蚀掉最后 100μm 到埋层氧化层。埋层氧化相对于硅来说有大于300:1 的刻蚀选择性，因此可以作为非常有效率的刻蚀停止层。这种方法的压倒性优势是，所有的硅能被均匀去除，留下一个非常平滑的（小于 10nm）表面用于薄层氧化物键合。而事实上，传送电路层自动成为有最小的跨越厚度的晶片，促进制造工艺流程中的高密度层间连接。因为层间的距离比传统的硅通孔方案的距离小很多，通孔的间距和大小极大地减小了。一个互连通孔的最小高度可以低到 1 ~ 2μm，允许硅通孔的直径小到 0.2 ~ 0.25μm[13]。如果极其严密的晶圆级对准也可以实现，1 ~ 2μm 量级的间距通孔是可以实现的，相反在宽松的硅通孔间距下，不可能实现新的系统级设计。图 2-4b 和 c 是传统的硅通孔和使用基于 SOI 的集成方案的铜填充硅通孔的一组对比。尺寸上的比较显示了基于 SOI 的 3D 集成电路制造的潜在优势。

2.3.5　通孔工艺集成和通孔类型的重新分类

由于通孔工艺集成方法必须符合具体的设计规则，用于制造硅通孔的具体工艺流程对于电路设计者来说是重要的。例如，对于某些通孔，其工艺本身需要在流水线后端排除规则来允许硅通孔穿过，而有些则不需要。另一个例子，正面一侧和背面一侧的通孔工艺的容差是不同的，这决定了具体的通孔工艺设计规则。在许多早期的文献中，术语“先通孔”和“后通孔”的使用已普遍，但这些术语都导致在

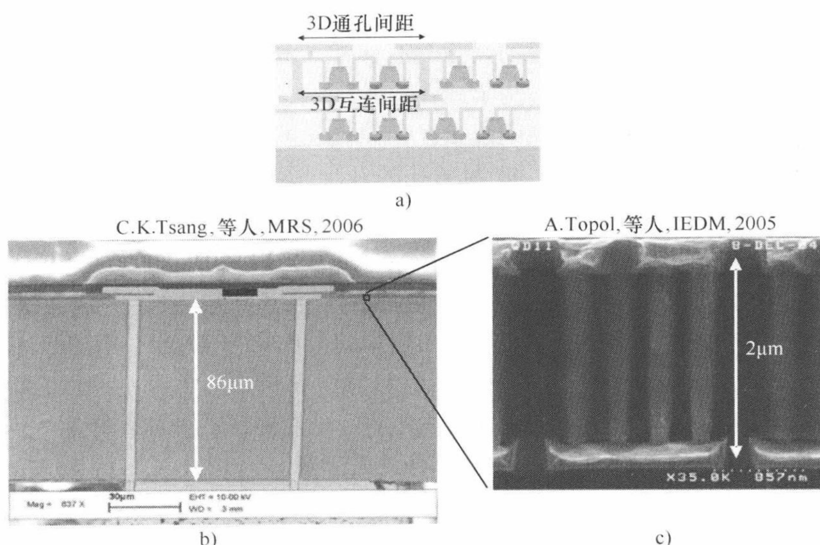


图2-4 a）基于SOI的面到背3D集成示意图；b）相对传统的硅厚度的环状硅通孔的尺寸；c）用SOI-3D技术的层间硅通孔尺寸的对比

工业应用中显著的混乱。有时这些指定用于表示通孔制作在基本集成电路晶圆完成之前或之后（例如，对于在晶圆正面加工通孔的工艺，这些术语用于区分在后端布线之前或之后形成）。在其他情况下，这些术语代表在完整的基本晶圆形成之前形成还是在晶圆减薄之后形成（例如从减薄的晶圆的正面或者背面）。一种可以代替的基于工艺流程中通孔刻蚀时间的分类方案可用来提高明晰度。这个重新定义的分类是基于两个最重要的对于设计的实际考虑：①这个工艺使用的是前面通孔刻蚀还是后面通孔刻蚀？②如果是前面通孔刻蚀，刻蚀被完成在后线布线之前还是之后？这些使我们定义了3个主要的关注点：

1）后端前正面通孔（F1类型）；

2）后端后正面通孔（F2类型）；

3）背面通孔刻蚀（衬底减薄后，B类型）。

由于背面蚀刻一般在最下面的后线金属层，我们忽略了这些情况，在这种情况下在后线工艺中，背面通孔刻蚀得更深。3种主要的硅通孔的分类大致情况见表2-1。

表2-1 不同硅通孔类型的特性

通孔类型	F1	F2	B
通孔刻蚀	正面	正面	背面
通孔形成	在BEOL之前	在BEOL之后	在减薄之后
高温材料兼容性	+	−	−
减小的通孔维度	+	−	+
低电路阻碍	+	−	+
工艺集成的方便程度	−	+	+

2.3.5.1 后端前正面通孔（F1 类型）

一种已经在 IBM 公司和其他地方验证的硅通孔方法是先通孔工艺流程。用这种方法，F1 通孔在后线工艺之前形成，并且这种方法有允许更高温度电介质和金属填充工艺，并具有高的深宽比的通孔形成的优点。F1 通孔在和传统的 CMOS 集成方面更具有挑战性，但是在后线中有低线阻的优点。一种特别的已经被描述的方法是环形通孔几何形状大面积的接触方法[17]。这种结构利用一种环形通孔刻蚀，其中环形的宽度足够窄，从而利用多种材料完整填充，包括多晶硅掺杂、铜电镀或者钨化学气相淀积。对于更高密度的 3D 集成应用，具有小的中心核的环体（该区域通过所述环的内径定义）可以被使用，其中环形的区域被绝缘电介质填充并且中心核接着刻蚀和金属化。“后端前侧通孔”可以扩展到在集成化之前立即发生的（例如，预 M1 前侧通过，经由预 M2 前侧）任何特定金属层级的所有情况。它把 F2 通孔和在 F2 通孔加工之后的 CMOS 制造完全分开。

2.3.5.2 后端后正面通孔（F2 类型）

硅通孔可以从晶圆的前面被刻蚀，利用前面对准能力，并且还仍能在后线布线完成之后被制造。这种 F2 工艺在其他许多情况下更容易和 CMOS 工艺技术集成，并具有不干扰标准 CMOS 制造工艺流程的优点。它具有通孔略大的不利方面，同时由于是通过后线延伸的通孔，也具有阻断布线通道的显著缺点。由于这个原因，除非通孔尺寸或所需的硅通孔的总数很小，否则这种通孔使用价值很小，特别是对于高复杂度的设计。

2.3.5.3 背面通孔刻蚀（衬底减薄后，B 类型）

各种垂直穿透硅互连技术已由 IBM 公司[17-19]开发。通常，硅通孔旨在利用一种背部深反应离子刻蚀的晶圆减薄之后形成。像这种 F2 通孔，B 类型通孔也有显著的优势，即在基本晶圆中使用的 CMOS 技术不需要修改。在背部深反应离子刻蚀之后，绝缘层可随后施加到通孔的内部并从底部选择性地除去允许到晶圆的电接触。用这种方式准备的金属化大通孔已经用多种方式说明。例如，初始部分铜填充，镀在铬/铜 BLM 和铅/锡焊料蒸发后已被验证。对于介质层和金属填充，背面通孔的纵横比通常有深宽比的限制。它们也将被显著减薄的晶圆背面光刻对准能力阻碍，从而可能不适合于高密度的 3D 集成。

2.4 总结

为了成为一个有效率的 3D 电路设计者，理解 3D 技术的工艺考量是重要的。在这一章中，我们尽力描述了 3D 电路设计者需要的基本的工艺考量，如面向层间的、层间对准的、界面压接设计、硅通孔的维度和 CMOS 集成工艺。这些考虑都直接影响设计，在一个特定的 3D 工艺中，3D 工艺的选择和电路优化都是十分重

要的。

在 CMOS 图像传感、晶圆级芯片级封装、存储器的封装、硅通孔、芯片-芯片和面积-阵列的连接等领域的技术发展具有很好的发展前景，并且全 3D 集成电路的发展，正在快速地推动我们不断前进。我们希望电路设计者读这本书时能得到灵感来学习如何更好地利用独特的 3D 集成。

致谢

作者想感谢以下对于本书所做出贡献的人：Roy R. Yu, Sampath Purushothaman, Kuan-neng Chen, Douglas C. La Tulipe, Narender Rana, Leathen Shi, Matthew R. Wordeman, Edmund J. Sprogis, Fei Liu, Steven Steen, Cornelia Tsang, Paul Andry, David Frank, Jyotica Patel, James Vichiconti, Deborah Neumayer, Robert Trzcinski, Latha Ramakrishnan, James Tornello, Michael Lofaro, Gil Singco, John Ott, David DiMilia, William Price, Jesus Acevedo。作者也感谢 IBM 公司微电子研究实验室，科学服务中心和 EV 组、Suss MicroTec 成员的支持。该项目部分由 DARPA SPAWAR 资助，合同号为 N66001-00-C-8003 和 N66001-04-C-8032。

参考文献

1. G. Moore, Cramming more components onto integrated circuits, *Electronics* 38, 114–117 (1965).
2. R. H. Dennard, F. H. Gaensslen, H. N. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, Design of ion-implanted MOSFETs with very small physical dimensions, *IEEE Journal of Solid-State Circuits*, SC-9, 256–268 (1974).
3. A. W. Topol, D. C. La Tulipe, L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini, and M. Ieong, Three-dimensional integrated circuits, *IBM Journal of Research & Development*, 504: 91–506 (2006).
4. C. Narayan, S. Purushothaman, F. Doany, and A. Deutsch, Thin film transfer process for low cost MCM-D fabrication, *IEEE Transaction Transactions on Components, Packaging, and Manufacturing Technology*, B18: 42–46 (1995).
5. E. D. Perfecto, R. R. Shields, A. K. Malhotra, M. P. Jeanneret, D. C. McHerron, and G. A. Katopis, MCM-D/C packaging solution for IBM latest S/390 servers, *IEEE Transaction on Advance Packaging*, 23: 515–520 (2000).
6. S. E. Steen, D. C. La Tulipe, A. Topol, D. J. Frank, K. Belote, and D. Posillico, Wafer scale 3D integration: overlay as the key to drive potential, *Microelectronic Engineering*, 84: 1412–1415 (2007).
7. A. W. Topol, D. C. La Tulipe, L. Shi, S. M. Alam, A. M. Young, D. J. Frank, S. E. Steen, J. Vichiconti, D. Posillico, D. M. Canaperi, S. Medd, R. A. Conti, S. Goma, D. Dimilia, C. Wang, L. Deligianni, M. A. Cobb, K. Jenkins, A. Kumar, K. T. Kwietniak, M. Robson, G. W. Gibson, C. D'Emic, E. Nowak, R. Joshi, K. W. Guarini, and M. Ieong, Assembly technology for three dimensional integrated circuits, *Proceedings of 22nd International VLSI Multilevel Interconnection Conference (VMIC)*, pp. 83–88, Fremont CA, October 4–6, (2005).
8. K.-N. Chen, C. S. Tan, A. Fan, and R. Reif, Morphology and bond strength of copper wafer bonding, *Electrochemical and Solid-State Letters*, 7: G14–G16 (2004).

9. K.-N. Chen, C. K. Tsang, A. W. Topol, S. H. Lee, B. K. Furman, D. L. Rath, J.-Q. Lu, A. M. Young, S. Purushothaman, and W. Haensch, Improved manufacturability of Cu bond pads and implementation of seal design in 3D integrated circuits and packages, *23rd International VLSI Multilevel Interconnection (VMIC) Conference,* Fremont CA, September 25–28, 2006, VMIC Catalog No 06 IMIC-050, pp. 195–202 (2006).
10. K.-N. Chen. S. H. Lee, P. S. Andry, C. K. Tsang, A. W. Topol, Y. M. Lin, J. Q.Lu, A. M. Young, M. Ieong, and W. Haensch, Structure design and process control for Cu bonded interconnects in 3D integrated circuits, *IEDM Technical Digest*, 20–22 (2006).
11. H. B. Pogge, et al., Bridging the chip/package process divide, *Proceedings of AMC*, 129–136 (2001).
12. R. Yu, Wafer level 3D integration, *International VLSI Multilevel Interconnection (VMIC) Conference*, Fremont CA, September 24–27, (2007).
13. A. W. Topol, D. C. La Tulipe, L. Shi, S. M. Alam, D. J. Frank, S. E. Steen, J. Vichiconti, D. Posillico, M. Cobb, S. Medd, J. Patel, S. Goma, D. DiMilia, T. M. Robson, E. Duch, M. Farinelli, C. Wang, R. A. Conti, D. M. Canaperi, L. Deligianni, A. Kumar, K. T. Kwietniak, C. D'Emic, J. Ott, A. M. Young, K. W. Guarini, and M. Ieong, Enabling SOI based assembly technology for three-dimensional (3D) integrated circuits (ICs), *IEDM Technical Digest*, 363–366 (2005).
14. P. Leduc, F. de Crécy, M. Fayolle, B. Charlet, T. Enot, M. Zussy, B. Jones, J.-C. Barbe, N. Kernevez, N. Sillon, S. Maitrejean, D. Louis, and G. Passemard, Challenges for 3D IC integration: bonding quality and thermal management, *Proceedings of IEEE International Interconnect Technology Conference (IITC)*, pp. 210–212 (2007).
15. A. W. Topol, S. J. Koester, D. C. La Tulipe, and A. M. Young, 3D fabrication options for high performance CMOS technology, *Wafer Level 3D ICs Process Technology*, C. S. Tan, R. J. Gutmann, and L. R. Reif, Eds., Springer, New York; ISBN 978-0-387-76532-7 (2008).
16. K. W. Guarini, et al., Process technologies for three dimensional integration, *Proceedings of the 6th Annual International Conference on Microelectronics and Interfaces,*American Vacuum Society, pp. 212–214 (2005).
17. C. K. Tsang, P. S. Andry, E. J. Sprogis, C. S. Patel, B. C. Webb, D. G. Manzer, and J. U. Knickerbocker, CMOS-compatible through silicon vias for 3D process integration, *Proceedings of Material Research Society*, 970: 145–153 (2006).
18. P. S. Andry, C. Tsang, E. J. Sprogis, C. Patel, S. L. Wright, and B. C. Webb, A CMOS-compatible process for fabricating electrical through-vias in silicon, *Proceedings of the 56th Electronic Components and Technology Conference*, San Diego, CA, pp. 831–837 (2006).
19. C. S. Patel, C. K. Tsang, C. Schuster, F. E. Doany, H. Nyikal, C. W. Baks, R. Budd, L. P. Buchwalter, P. S. Andry, D. F. Canaperi, D. C. Edelstein, R. Horton, J. U. Knickerbocker, T. Krywanczyk, Y. H. Kwark, K. T. Kwietniak, J. H. Magerlein, J. Rosner, and E. J. Sprogis, Silicon carrier with deep through vias, fine pitch wiring and through cavity for parallel optical transceiver, *Proceedings of the 55th Electronic Components and Technology Conference*, pp. 1318–1324 (2005).

第3章　三维（3D）芯片的热和电源传输挑战

Pulkit Jain　Pingqiang Zhou　Chris H. Kim　Sachin S. Sapatnekar

与2D集成电路相应部分对比，3D集成电路为每单位英尺集成度的迅速增长提供了潜在的可能。虽然这种能力对大量应用场合很有吸引力，但也在热管理和电源传输领域创造出了更难突破的设计瓶颈。首先，随着集成度的提高，单位英尺的热量增加会导致潜在的片上温度升高。热管理的任务必须同时通过热沉和使用基于热的设计方法共同实现。这里，热沉是指将内部产生的热传导到外部。第二，封装引脚需传输到3D芯片中的电源量大量增加，导致了可靠的电源传输异常复杂。本章给出了这些问题的整体总结，并简述了解决相关瓶颈问题的方案。

3.1　介绍

3D芯片的主要优点源于它们有比2D电路更高密度地将电路集成的能力。然而，增加了的集成度也导致了副作用，其表现在对设计者有新的限制和挑战。热和电源传输问题均可追溯到这样的事实，即在基本相同的封装工艺下，k堆叠层3D芯片会使用k倍单个同等尺寸2D芯片所需的电流。其影响如下：

1）3D芯片产生k倍于2D芯片的功耗，意味着其对应产生的热量必须被送到外部环境中。如果设计技术不考虑热并且2D和3D电路的封装的热特性相似，那么3D芯片的片上温度将远高于2D芯片。直线上升的温度会使芯片性能具有不可预测性，并同时损害性能和可靠性。因此，片上热管理是3D设计的关键问题。

2）封装必须能够通过电源供电（V_{dd}和地）引脚提供k倍于2D芯片的电流。此外，由于硅通孔（TSV）给供电网络造成附加电阻，电源传输问题在3D IC中更加严峻。要知道可靠的电源网格设计即使在2D设计中也是主要的瓶颈。这说明需要投入大量的资源来构建3D芯片可靠的电源网格。

本章给出了关于热和电源网格设计问题的概述。我们首先在3.2节中聚焦热问题，给出热分析方法总结，随后是热优化技术的指示词，其细节在本书的其他几个章节中有所强调。之后，在3.3节中，我们描述了3D系统电源传输的挑战和克服它们的解决方案。

3.2　三维集成电路中的热问题

3.2.1　热 PDE

全芯片热分析包括经典热传导理论的应用。其区别在于片上背景下特定的混合问题。例如，片上形状本质上是直线型的并含有矩形形状对称性；器件作为主要热源，存在于每个 3D 堆叠层的每一个单独的层中，而用户通常对分析器件层之中的温度感兴趣。

芯片中传统的热传导用傅里叶传导法则描述[29]，其声明热流量 $q(\mathrm{W/m^2})$ 与温度 T（K）的逆向梯度成正比，比例系数为对应材料的热传导系数 $k_t[\mathrm{W/(m \cdot K)}]$，即

$$q = -k_t \nabla T \tag{3-1}$$

q 在一个区域中的散度是生成的功耗与区域中热能随时间的变化率的差值，即

$$\nabla \cdot q = -k_t \nabla \cdot \nabla T = -k_t \nabla^2 T = g(r,t) - \rho c_p \frac{\partial T(r,t)}{\partial t} \tag{3-2}$$

式中，r 是确定温度的点的空间坐标；t 代表时间（单位：s）；g 是单位体积功耗密度（单位为 $\mathrm{W/m^3}$）；c_p 是芯片材料的热容量（单位：$\mathrm{J/(kg \cdot K)}$）；ρ 是材料密度（单位：$\mathrm{kg/m^3}$）。该式也可以写成如下的热方程形式，即二次偏微分方程（PED）形式

$$\rho c_p \frac{\partial T(r,t)}{\partial t} = k_t \nabla^2 T + g(r,t) \tag{3-3}$$

热导率 k_t 在均匀介质中是各向同性的。硅、二氧化硅和金属（如铝和铜）的热导率是基础材料性能，它们的数值可以查表得到。实际上，在分析的早期阶段，为了优化目的，集成电路可假定是层级热导率均匀的。体层具有体硅的热导率，金属层的热导率通常用平均方式计算得到。这是由于该区域由二氧化硅和金属混合得到，那么根据区域中的金属密度，可使用等效热导率进行宏观分析。

式（3-3）的解与瞬态热响应有关。在稳态下，所有的与时间相关的微分值趋近于 0，因此，该偏微分方程（即 PED）的稳态分析解为

$$\nabla^2 T(r) = -\frac{g(r)}{k_t} \tag{3-4}$$

即为众所周知的泊松方程。

热传导的时间常数是百万秒量级的，远远长于当今 VLSI 电路中的亚纳秒量级时钟周期。因此，如果电路在同等功耗模式下保持更长的时间周期并且其功耗密度分布保持相对稳定，那么稳态分析可以准确地获取该电路的热行为。即使实际情况有所不同，稳态分析仍对早期和较粗略的分析非常有用。同理，稳态分析用于在设

计早期阶段分析电源网格网络。另一方面，当有更细致层次的输入或电路以高于热时间常数的时间间隔多次改变功耗模式时，瞬态分析则成为可能的和潜在的有效方法。

为获得式（3-3）良好定义的解，必须设定一系列边界条件。典型地，在芯片层级的边界条件包括构建宏模型包，并假定这些宏模型与恒定的周围环境温度互相作用。

3.2.2 稳态热分析算法

我们下面将基于有限差分法（FDM）和有限元法（FEM）的应用描述稳态分析技术。两种方法均将整个芯片离散并形成关于芯片中温度分布到功耗密度分布的线性方程系统。FDM 和 FEM 的主要区别是 FDM 离散差分操作数和 FEM 离散温度场。FDM 和 FEM 的主要优点是它们能处理复杂材料结构，特别是处理 VLSI 芯片中不均匀互连分布。

FDM 和 FEM 都会产生问题公式，这些问题公式需要巨大的线性方程系统的解。描述这些方程的矩阵基本上是稀疏矩阵（FDM 中比 FEM 更多，可从独立元素标记 stamp 中看出）并确定是正矩阵。有许多不同的方法来解这些方程[9]。直接法通常使用高斯变量消去法，如 LU 分解，来首先分解矩阵。然后通过前向和后向减法求解整个系统。对于 $n\times n$ 密集矩阵，LU 分解的代价是 $O(n^3)$ 的运算复杂度，但实际上对于稀疏系统只是略超过线性而已。随后是前向/后向减法步骤，对每行输入数受常量限制的稀疏系统执行 $O(n)$ 次。如果一个系统是对大量右手向量求值，对应不同的功耗向量，那么 LU 只需执行一次，并且其代价可由多输入向量的解平摊。

迭代的方式对大稀疏正有限矩阵非常有效。这类技术包括更多类似高斯—雅可比法（Gauss-Jacobi），高斯—赛德尔法（Gauss-Seidel）的经典方法及逐次超松弛迭代，也包括更多的基于共轭梯度法或 GMRES 法的当代方法。这里的思路从初始假设解开始并接着提炼它以实现收敛。在特定情况下可以保证收敛：特别的，FDM 矩阵具有专门保证该结果的结构。对于 FDM 方法，电源矩阵分析问题的相似性引入了相似的解决技术，包括随机漫步方法[30,31]和其他如多栅格方法[23,24]的技术。

3.2.2.1 FDM 方程的公式推导

稳态泊松方程式（3-4）可以通过写出温度的二阶空间偏导数分解。其中温度 T 是直角坐标的有限差。空间区域可以分解成多个长方形，每个由一个节点以及分别在 x、y、z 方向上长 Δx、Δy、Δz 长度的面代表。假设我们感兴趣的区域放置在第一个八分区内，顶点处于原点。我们将使用 $T_{i,j,k}$ 来代表节点（$i\Delta x$，$j\Delta y$，$k\Delta z$）的稳态温度，并且芯片中的每一个节点都配有一个等式。

该划分可用于得到空间偏导数的估计值。例如，在 x 轴方向上，可以得到

$$\frac{\partial^2 T(r)}{\partial^2 x} \approx \frac{\frac{T_{i+1,j,k}-T_{i,j,k}}{\Delta x}-\frac{T_{i,j,k}-T_{i-1,j,k}}{\Delta x}}{\Delta x} \tag{3-5}$$

$$= \frac{T_{i+1,j,k}-2T_{i,j,k}+T_{i-1,j,k}}{(\Delta x)^2} \tag{3-6}$$

在 y 和 z 轴方向上可以得到类似的等式

定义算子 δ_x^2，δ_y^2，δ_z^2 为

$$\begin{aligned}
\delta_x^2 T_{i,j,k} &= T_{i+1,j,k}-2T_{i,j,k}+T_{i-1,j,k} \\
\delta_y^2 T_{i,j,k} &= T_{i,j+1,k}-2T_{i,j,k}+T_{i,j-1,k} \\
\delta_z^2 T_{i,j,k} &= T_{i,j,k+1}-2T_{i,j,k}+T_{i,j,k-1}
\end{aligned} \tag{3-7}$$

使用有限差分进行泊松方程的 FDM 分解得到如下的线性方程系统：

$$\frac{\delta_x^2 T_{i,j,k}}{(\Delta x)^2}+\frac{\delta_y^2 T_{i,j,k}}{(\Delta y)^2}+\frac{\delta_z^2 T_{i,j,k}}{(\Delta z)^2} = -\frac{g_{i,j,k}}{k_t} \tag{3-8}$$

特别是对于电气工程人员，分解过程更好的可视化采用了另一种热传导理论中的标准设备。此设备通过所谓的热点模拟构架等效热电路。这里分解的每个节点对应电路的一个节点。稳态方程对应“热阻”在节点和“热电流源”间连接的网络。其中节点对应空间相邻区域。“热电流源”映射到电源上。于是该热电路中节点的电压可以通过求解电路计算得到，并且由此得到这些节点的温度。数学上，可以从式（3-4）以稍微不同的方式写出分解后等式的方法得到式（3-8）。例如，在 x 轴方向，式（3-5）中的有限差分可写为

$$\frac{\partial^2 T(r)}{\partial^2 x} \approx \left[\frac{T_{i+1,j,k}-T_{i,j,k}}{R_{i+1,j,k}}-\frac{T_{i,j,k}-T_{i-1,j,k}}{R_{i-1,j,k}}\right]\frac{1}{k_t A_x \Delta x} \tag{3-9}$$

式中，$R_{i-1,j,k}=\frac{\Delta x}{k_t A_x}$，并且 $A_x=\Delta y \Delta z$，是元件沿 x 轴切开时纵切面的面积。由此得到以下的分解：

$$\left[\frac{T_{i+1,j,k}-T_{i,j,k}}{R_{i+1,j,k}}+\frac{T_{i-1,j,k}-T_{i,j,k}}{R_{i-1,j,k}}\right]+\left[\frac{T_{i,j+1,k}-T_{i,j,k}}{R_{i,j+1,k}}+\frac{T_{i,j-1,k}-T_{i,j,k}}{R_{i,j-1,k}}\right]+$$
$$\left[\frac{T_{i,j,k+1}-T_{i,j,k}}{R_{i,j,k+1}}+\frac{T_{i,j,k-1}-T_{i,j,k}}{R_{i,j,k-1}}\right] = -G_{i,j,k} \tag{3-10}$$

式中，$G_{i,j,k}=g_{i,j,k}\Delta V$ 是元件中产生的总功耗，$\Delta V = A_x \Delta x = A_y \Delta y = A_z \Delta z$。

式（3-10）可视为与每个电路节点的节点方程等效，其中该节点连接到对应其通过热阻相连的 6 个相邻元件的节点，如图 3-1 所示。或者说，使用 FDM 的热分析方程的解相当于拥有线性电阻和电流源的电路的解。

电路的接地节点或参考节点对应一个温度恒定的节点，通常是环境温度。如果使用绝热边界条件，这就意味着这些节点连接到了对应接地节点的周围环境。另一方面，对封装和散热器使用一个更细致的由热阻和热容的互连结构组成的热模型是

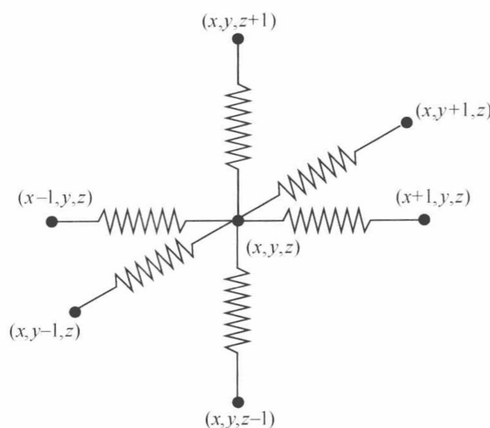


图 3-1 在 FDM 离散化后连接到节点（x，y，z）的热阻

可能的，如在 HotSpot[1,38] 中使用的模型，也可以使用另外一种紧凑的降阶模型。在两种情况下，一个或多个封装模型的节点都将连接到周围环境，即认定的接地节点。这样的模型获得可通过在封装上应用（例如 FDM 或 FEM）并提取（可能稀疏化了的）宏模型，此模型保持了将封装连接到芯片和周围环境的端口。

电路的总方程可通过改良的节点分析法[7] 得到，我们可进一步获得一系列方程：

$$\boldsymbol{G}\boldsymbol{I} = P \tag{3-11}$$

式中，$\boldsymbol{G}$ 是 $n \times n$ 矩阵；T 和 P 是 n 元向量，其中 n 对应电路中的节点数。容易证明 $\boldsymbol{G}$ 矩阵是稀疏导数矩阵，具有带状结构，对称，并且对角占优。

对于瞬态热分析，式（3-3）中时间相关的左侧项非零。使用与上述方法类似的有限差分，该式可在空间领域分解为

$$\rho c_p \frac{\partial T_{i,j,k}}{\partial t} = k_t \left[\frac{\delta_x^2 T_{i,j,k}^{n+1} + \delta_x^2 T_{i,j,k}^{n}}{2(\Delta x)^2} + \frac{\delta_y^2 T_{i,j,k}^{n+1} + \delta_y^2 T_{i,j,k}^{n}}{2(\Delta y)^2} + \frac{\delta_z^2 T_{i,j,k}^{n+1} + \delta_z^2 T_{i,j,k}^{n}}{2(\Delta z)^2} \right] + \frac{g_{i,j,k}^{n}}{\rho c_p} \tag{3-12}$$

右侧的时间相关项也可视为通过单位体积热阻的电流和热电流源。另一方面，左侧代表值为 $\rho c_p \frac{\partial T_{i,j,k}}{\partial t}$的电流源。回想在热电模拟学中，温度对应电压，容易理解我们能用单位体积容值为 ρc_p的热容代表左侧项。

根据此映射关系，通过创建由电阻、电流和电容组成等效网络并使用瞬态分析的常规电气技术，即可实现瞬态热分析。

3.2.3 有限元法（FEM）

FEM 提供了另一条解式（3-4）形式的泊松方程的道路。虽然这是一种用于解

此类 PDE 方程的基本的、经典的、并被广泛使用的技术，但仍可使用片上问题的参数（见 3.2.1 节开始处）来有效的计算解。

在本章参考文献［10］中提供了当应用在片上情况时的一个 FEM 的简要解释。在有限元分析中，设计空间首先分割或网格化成元素。可以使用如四面体和六面体的不同的元素形状。对于片上问题，所有的热源均建模为长方形，是利用 FEM 把芯片分成 8 节点直角六面体元素的一种合理分割[11]，如图 3-2 所示。在片上情况下，六面体元素还简化了 FEM 过程中预订—保持和数据管理的问题。元素节点的温度为未知数，由有限元分析计算得到。元素内部的温度的计算使用插值函数来估计元素内部热方程的解，如式（3-13）所示。

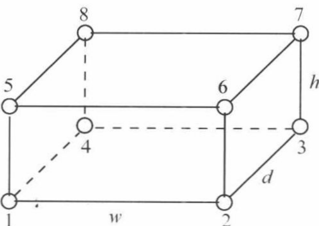


图 3-2　一个 FEM 中使用的 8 节点六面体单元

$$T(x,y,z) = \sum_{i=1}^{8} N_i(x,y,z)T_i \tag{3-13}$$

式中，$N_i(x, y, z)$ 是节点 i 的形状函数，T_i 是节点 i 的温度。让 (x_c, y_c, z_c) 作为元素的中心，并用 w、d、y 分别代表元素宽度、深度和高度。元素中任意点的温度使用形状函数在 FEM 中插值，此时 $N_i(x, y, z)$ 可写为如下的三线函数：

$$N_i(x,y,z) = \left(\frac{1}{2} + \frac{2(x_i - x_c)}{w^2}(x_i - x_c)\right) \times \left(\frac{1}{2} + \frac{2(y_i - y_c)}{d^2}(y_i - y_c)\right) \times \left(\frac{1}{2} + \frac{2(z_i - z_c)}{h^2}(z_i - z_c)\right) \tag{3-14}$$

该函数的特性是其值在原点 i 为 1，在所有其他顶点为 0。这满足了对应原点温度的基本要求，如式（3-13）计算的那样。

根据形状函数，热梯度 g 可以使用式（3-13）得到，如下：

$$\boldsymbol{g} = \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{bmatrix} = \boldsymbol{B}T \tag{3-15}$$

式中

$$\boldsymbol{B} = \begin{bmatrix} \frac{\delta N_1}{\delta x} & \frac{\delta N_2}{\delta x} & \cdots & \frac{\delta N_8}{\delta x} \\ \frac{\delta N_1}{\delta y} & \frac{\delta N_2}{\delta y} & \cdots & \frac{\delta N_8}{\delta y} \\ \frac{\delta N_1}{\delta z} & \frac{\delta N_2}{\delta z} & \cdots & \frac{\delta N_8}{\delta z} \end{bmatrix} \tag{3-16}$$

如在使用改进的节点公式的电路仿真的例子里，为每个元素创建标志并添加到全局系统方程中，得到

$$\boldsymbol{K}_g T = P \tag{3-17}$$

式中，T 是所有节点温度的向量。该系统方程是标准稀疏的并可高效求解。

在 FEM 中，这些标志为元素硬度矩阵 $\boldsymbol{K}$，其值可通过使用基于变分法的技术确定。由于该理论的完整引用超出了本章的范围并可在 FEM 的标准文本（如本章参考文献［25］）中找到，注意到最终结果导出了后续标志就足够了。由于本例中只考虑热传导，于是有

$$\boldsymbol{K} = \int_V B^{\mathrm{T}} \boldsymbol{D} B \mathrm{d}V \tag{3-18}$$

式中，V 是元素的体积；$\boldsymbol{D} = \mathrm{diag}(k_{t,x}, k_{t,y}, k_{t,z})$ 是 3×3 对角矩阵，其中 $k_{t,i}$，$i \in \{x, y, z\}$。其在沿 3 个坐标轴方向各向异性的情况下，代表在 3 个直角坐标方向上的热导率。在许多情况下，$k_{t,x} = k_{t,y} = k_{t,z} = k_t$。

对于我们的六面体元素，传导情况下的标志通过 8×8 的对称矩阵给出。该矩阵的输入仅依赖 w、h 和 d，具体如下：

$$\boldsymbol{K} = \begin{bmatrix} A & B & C & D & E & F & G & H \\ B & A & D & C & F & E & H & G \\ C & D & A & B & G & H & E & F \\ D & C & B & A & H & G & F & E \\ E & F & G & H & A & B & C & D \\ F & E & H & G & B & A & D & C \\ G & H & E & F & C & D & A & B \\ H & G & F & E & D & C & B & A \end{bmatrix} \tag{3-19}$$

式中

$$A = \frac{k_{t,x}hd}{9w} + \frac{k_{t,y}wd}{9h} + \frac{k_{t,z}wh}{9d}, B = -\frac{k_{t,x}hd}{9w} + \frac{k_{t,y}wd}{18h} + \frac{k_{t,z}wh}{18d}$$

$$C = -\frac{k_{t,x}hd}{18w} - \frac{k_{t,y}wd}{18h} + \frac{k_{t,z}wh}{36d}, D = \frac{k_{t,x}hd}{18w} - \frac{k_{t,y}wd}{9h} + \frac{k_{t,z}wh}{18d}$$

$$E = \frac{k_{t,x}hd}{18w} + \frac{k_{t,y}wd}{18h} - \frac{k_{t,z}wh}{9d}, F = -\frac{k_{t,x}hd}{18w} + \frac{k_{t,y}wd}{36h} - \frac{k_{t,z}wh}{18d}$$

$$G = -\frac{k_{t,x}hd}{36w} - \frac{k_{t,y}wd}{36h} - \frac{k_{t,z}wh}{36d}, H = \frac{k_{t,x}hd}{36w} - \frac{k_{t,y}wd}{18h} - \frac{k_{t,z}wh}{18d}$$

多个元素的标志，包括独立传导和对流标志，如果可应用的话还包括功耗向量，现在其可以叠加在一起以得到整体刚度矩阵。整个网格由这些六面体在网格中排列组成，每个节点最多由 8 个不同元素共享。通过把对应到相同节点的元素矩阵相加，元素硬度矩阵可规格化得到全局硬度矩阵 K_g。全局功耗向量的每个输入 P，包含功耗和对应节点生成的热，以及可能附加的对流元素。

所有这些标志都合并到全局公式系列中，见式（3-17）。在绝热边界条件下或一个节点连接到周围环境的情况下，对应温度应设定到周围环境温度。方程和变量的数量可相应减少。例如，如果 T_1 是未知的温度向量，并且子向量 T_2 的所有节点均连接到固定温度，那么全局硬度矩阵可写为以下形式：

$$\begin{bmatrix} K_{g,11} & K_{g,12} \\ K_{g,21} & K_{g,22} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \tag{3-20}$$

T_2 的固定值可以移到等式右侧以获得化简的公式，即

$$K_{g,11} T_1 = P_1 - K_{g,12} T_2 \tag{3-21}$$

3.2.4　三维电路热优化

图 3-3 展示了 3D 电路的简单热模型和克服这些结构中热挑战的技术梗概。图中示出了一个热沉上的 3D 芯片的结构体，该模型使用分布式电源网给分布式电阻网络供电，该电阻网连接到描述热沉的热阻模型上。虽然这是一个粗略的模型，但其满足说明的要求。通过运用热电模拟，该网络的电压代表了线片上的温度。温度由此可通过使用以下设计方法降低：

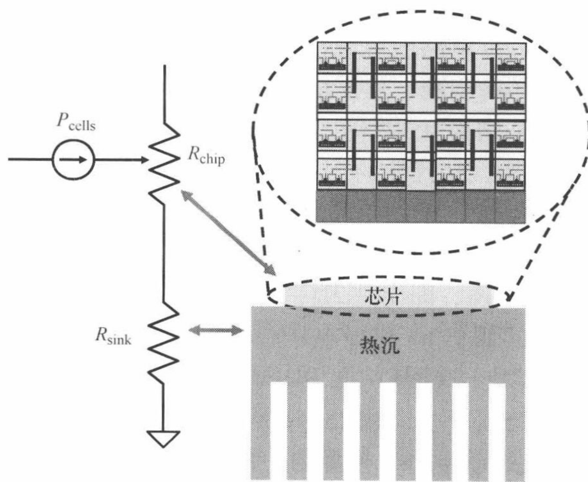


图 3-3　3D 芯片的简单热模型

1）低功耗设计：通过减小芯片的功耗，注入网络的热电流随之减小，控制了电压降（IR-drop），并因此控制了电压。

2）通过重新排布热源：热源的位置可以在物理设计（布局规划和布局）过程中调整来获得优化的温度。简单地讲，这意味着高功耗模块要相互远离并离热沉更近。

3）通过改进传热渠道：温度降低还可以通过改进从器件到热沉的路径的有效热导率实现。达到此目的的一种有效方法是通过热通孔的插入。热通孔结构上类似

电气通孔但不承担电气功能。它们的主要作用是将热从 3D 结构中传导出去并将其传送到热沉。

4）通过改善热沉：一个改善了的热沉将使得 R_{sink} 值减小，这将帮助降低温度。

我们将在本书的其他部分讨论这些技术，仅除了最后一种，因为它超出了本书的范围。

3.3　三维芯片中的电源传输

尽管最近3D 芯片的研究热潮，但在电路设计和自动化圈子里仅有很少关于3D 芯片电源传输的研究。由于电源供电网络（PSN）阻抗的按比例缩小没能跟上由于有限线网资源和单位线网长度恒定 RC[26] 造成的器件密度和操作电流的增加，片上电源供电噪声在现代系统中已经恶化。正如前面说明的，这种情况在 3D 芯片中更加恶化。在 3D 芯片中增加的 IR 和 *Ldi/dt* 供电噪声会引起其工作速度的更大变化，从而导致更多的时序违反。寄生电感引发的供电噪声过冲会使可靠性问题恶化，如氧化层击穿、热电子注入（HCI）和负偏压温度不稳定性（NBTI）[2,23]（其受升高的温度影响而变严重）。综上所述，片上电源传输将成为 3D 芯片的关键挑战。

本节以传统高性能 2D 电路的电源传输问题的基本概述开始。接着，在 3.3.2 节中研究 3D 芯片电源传输问题、建模技术以及和传统 2D 芯片的比较。3.3.3 节中说明了试图采用 3D 拓扑结构 IC 技术以减轻特定电源传输问题的几个可信的结构，随后说明了几种用于电源网格优化的 3D 特定 CAD 技术。

3.3.1　电源传输基础

根据按比例缩小路线图，即使是 2D 芯片而言，未来高性能芯片也将需要多个总电流超过 $100A/cm^2$ 的低于 1V 供电电压[4]。针对高性能芯片的传统电源传输方法采用了称为稳压器模块（VRM）的 DC-DC 转换器。VRM 通常安装在主板上，通过外部连接提供电源至芯片，如图 3-4a 所示。片内电源传输网络如图 3-4b 所示。图中示出了一个微处理器 PSN 模型的一部分[28]。由 I/O 引脚和绑定线引发的封装的寄生参数建模为电感和电阻的串联。图中的去耦电容用于抑制瞬态噪声并包括外部去耦电容和 MOS 栅极电容等多种电路元件的电容。

芯片作为分布式噪声源从不同位置和不同频率吸取电流，引起供电传输缺陷。到达处理器的供电受通过封装的 IR 和 *Ldi/dt* 下降的影响组成了供电噪声，封装阻抗很大程度上不受工艺按比例缩小影响。然而按比例缩小仍然导致了一些意想不到的片上影响，即从一个工艺节点到下一个工艺节点不断增加的电流和更快的瞬态转换。前者使电压降恶化，后者加剧了 *Ldi/dt* 降。在以上这些影响之上是全局谐振噪声，在或靠近其频率处供电阻抗更容易于出现较大降幅。伴随这些级别变高的噪声和随 V_{dd} 按比例缩小而减小的噪声容限，向大功耗芯片的可靠的电源传输成为主要的挑战。

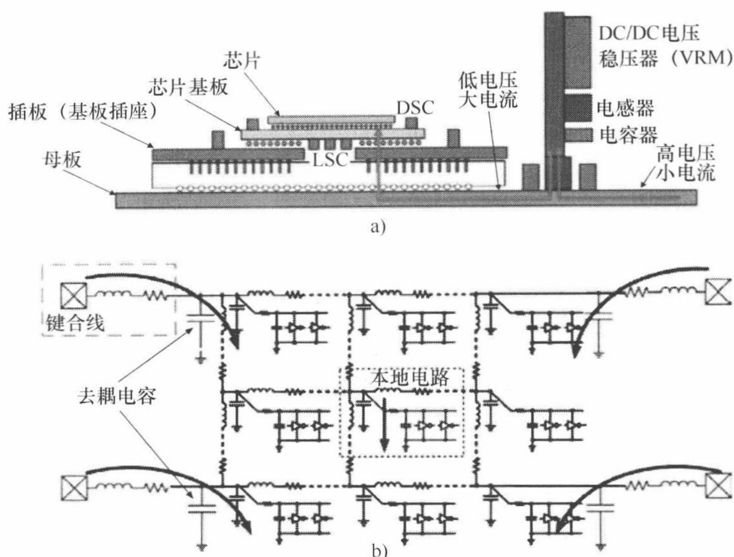


图3-4 a）传统电源传输结构[39] © 2007 IEEE；b）片上电源网格

典型电源网格的噪声谱如图3-5a所示。噪声的DC部分由穿过封装和电源网格的电压降决定。图中的第一个尖峰对应其谐振频率，即 $f_{res}=1/(2\pi\sqrt{LC})$，其一般出现在100～300MHz的范围内。在处理器循环操作或唤醒过程中会触发此频率的激励。可以在图中看到的另外几个尖峰源于时钟频率的开关动作及其高阶谐波，或者源于本地谐振：对应的噪声一般幅度上比本征谐振频率小一个数量级。图3-5b示出了一个独立测试结构的供电阻抗实测谱，其证实了图3-4所示的仿真模型。特定频率的噪声是通过在该频率的阻抗和电流分量的乘积估计的[15,41]。

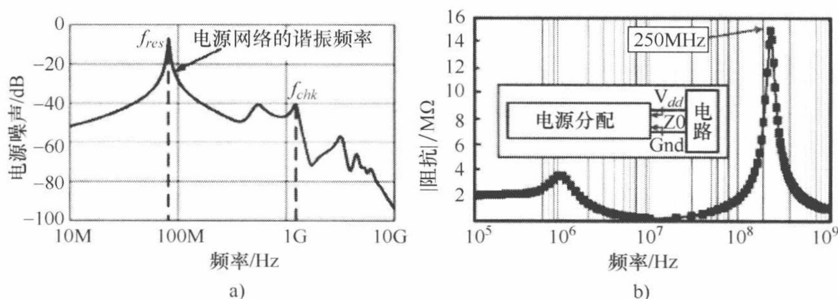
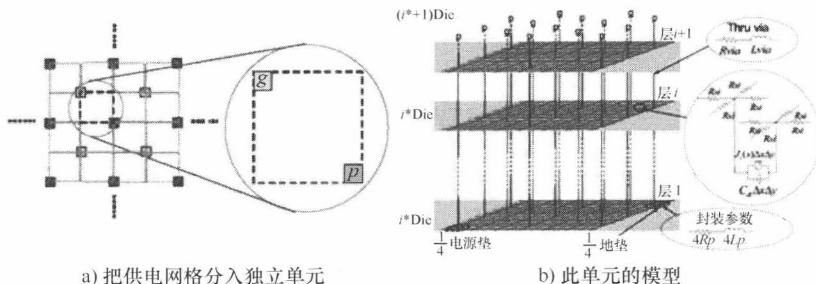


图3-5 a）供电噪声谱仿真；b）供电噪声实测结果[15] © 2006 IEEE

3.3.2 三维芯片电源传输：模型和挑战

在本节，我们关注3D芯片的电源传输问题并在此框架下分析PSN噪声问题。图3-6[19]中示出了一种基于片上和封装结合的电源供电结构的分布式模型的

3D 芯片模型。电源输入封装，通过分布在最底层芯片上的电源 I/O 压焊点，最后使用 TSV 到达其他上层芯片。芯片范围可以分成单元，即在一对相邻的电源和地引脚之间相同的方块区域，如图 3-6a 所示。在图 3-6b 中，这些单元以相邻 TSV 间几个子单元组成的网格的形式连接起来。从电学上看，每个 TSV 建模为电阻和电感组合的串联。平面方块单元采用集总式模型，其中的 R_{si}，J_i 和 C_{di} 分别代表网格电阻、有效电流密度和芯片单位偏压耦合电容。由于引脚压焊点由 4 个独立单元共享，封装参数以 1/4 归一化。子单元可多次重复以实现完整的 3D 功能模块。



a) 把供电网格分入独立单元　　b) 此单元的模型

图 3-6　3D IC 分布式模型[19]

电源网格模型必须与 3D 工艺相结合。图 3-7a 示出了 MIT 林肯实验室[5]的产品级 0.18μm 3D 工艺的 3D 芯片纵切面模型。该工艺包含 3 个芯片层。绑定压焊点在顶层，热沉一般在底层之下。处理器或其他功耗密集电路将被最理想地放置在底层紧靠热沉的位置。

为满足电和热传导，芯片层间通过 TSV 相连。图 3-7b 示出了连接了顶层背金属和底层顶金属的堆叠 TSV 的纵切面扫描电子显微镜（SEM）的图像。一个简化的电阻模型叠加在图上。基于实际参数提取[5]，每个堆叠的锥状 TSV 在该工艺下的电阻为 1Ω。顶层和中间层的正—反相对，而中间层和底层正—正相对，使顶层到中间层的路径更长，电阻更大。该配置模型可以通过将 1Ω 的通孔总电阻打断为 0.25Ω、0.5Ω 和 0.25Ω 的 3 段来实现，如图 3-7b 所示。由于在实验中发现 TSV 电感和电容的值相当小，故可以忽略[45]。

相对于传统 2D 的情况，供电路径中的 TSV 电阻对 3D 电源传输提出了新的挑战[20]。首先，由于 PSN 网络中增加的电阻，较低的芯片层承受了更多的 PSN 噪声。此外，由于功耗密集的电路必须放置在底层，使得可靠的电源传输更加困难。

相比传统 2D 芯片的模型，3D 情况下有两点重要的变更。首先，对于同样的电路，缩小了的芯片面积有效地增加了封装的寄生参数、由于供电引脚和绑定线的数量与供电电流的比率减小，封装电阻和电感的比重增加。其次，每芯片层噪声特征受供电路径上附加的 TSV 电阻影响。

图 3-8a 示出了已开发的用来比较 3D 和 2D 情况的电路模型。这些模型基于分布式供电网络模型的阻抗包络的曲线拟合，以及典型去耦电容和封装寄生参数值。

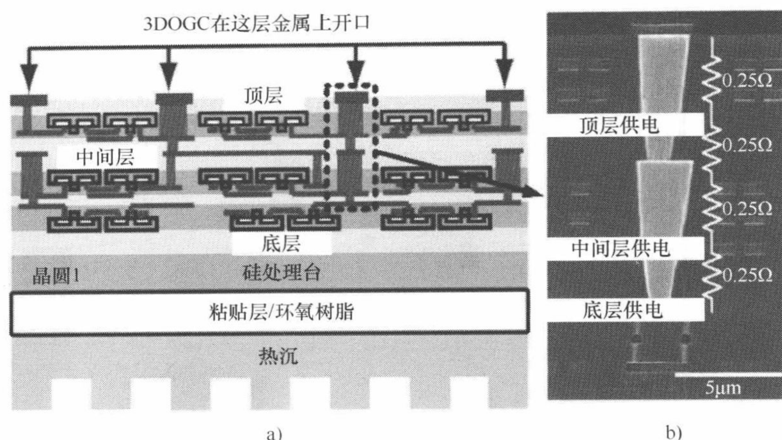


图 3-7　a）3D FD-SOI 工艺的纵剖面图；b）与纵剖面 SEM 图像对齐的简化通孔模型[21] © IEEE 2007

在 3D 中，我们发现供电路径将被 TSV 主导。相等尺寸中的芯片总电容（在 2D 情况下为 3nF）在 3D 芯片中将由 3 个芯片层等分。此外，由于 3D 芯片尺寸的缩小，电源引脚的数量会是 2D 情况下的 1/3，造成 3 倍封装寄生电感和电容值增加。

由于底层噪声可以预料是最差的，我们将该层的阻抗响应曲线与 2D 情况比较。归一化阻抗比较见图 3-8a，其说明了以下几点：

1）低频阻抗：在低频段，电容和电感分别是开路和短路电路。因此，2D 模型的电感值为 $2\times(0.01\Omega+0.03\Omega)=0.08\Omega$，而 3D 模型阻抗为 $2\times(0.03\Omega+0.05\Omega+0.1\Omega+0.05\Omega)=0.46\Omega$。这说明对于等量电流，3D 芯片将出现相对于 2D 芯片 $0.46/0.08=5.75$ 倍的电压降㊀。

2）谐振峰阻抗：谐振峰是由阻尼量和电感值决定的。其中，3D 电感增加的比重被落在底层的更大的电阻提供的阻尼抵消，尖峰显示了相对值。

3）谐振频率：二维电路谐振频率一般在 50～300MHz 之间，由 $f_{res}=1/(2\pi\sqrt{LC})$ 决定。如果 3D 的等效电容与我们的模型相同，受增加的 L 的影响，峰值会移动到如图 3-8c所示的更低的频率处。

4）高频阻抗：在高频段，由于底层电容的屏蔽效应，即电容在高频视为虚短电路的事实，2D 和 3D 阻抗趋向一致。

显然，相较于对应的 2D 设计，DC 供电噪声成为 3D 设计中更受关注的问题。

为了理解不同芯片层的供电噪声行为，我们通过分析仿真所得的跨不同芯片层的阻抗谱（见图 3-8c）得到 3D 芯片模型。关键结果如下：

㊀ 2R 原书误，应为 IR。——译者注

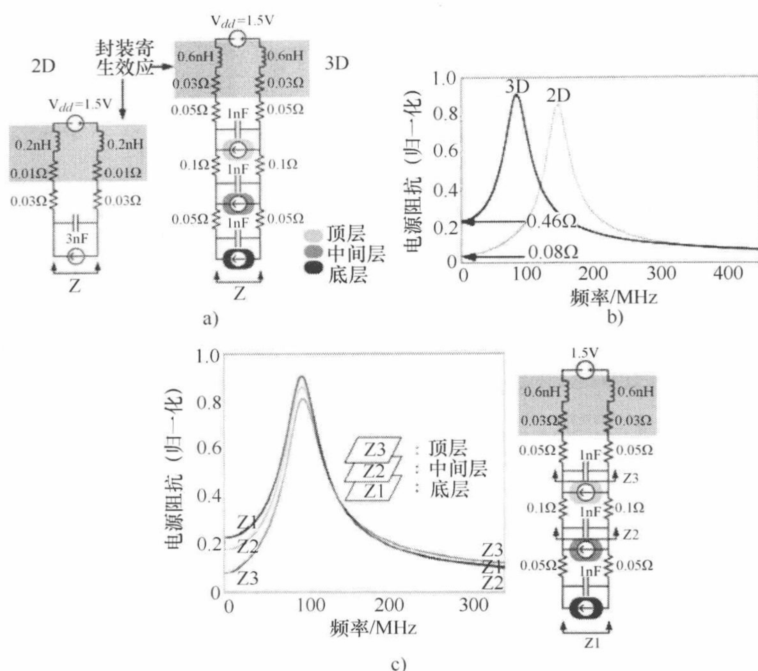


图 3-8　a）用于对比 2D 和 3D 阻抗响应的简化 PSN 模型；b）2D 和 3D 间阻抗响应比较；c）3D IC 所述 3 层的阻抗响应[20] © IEEE 2007

1）低频阻抗：正如所料，由 TSV 电阻决定的 DC 和低频阻抗在较低芯片显出恶化趋势。

2）高频阻抗：在高频段，顶层有最大阻抗，而中间层具有最小的 AC 阻抗。虽然这看上去与直观判断相反，但是可以解释为相邻层间电容的屏蔽/去耦效应，其造成了有效阻尼电阻在中间层最大且顶层最小。以上趋势在谐振峰以上的高频段更明显。

3）谐振行为：由于上述的屏蔽效应在中频段不明显，谐振峰延续落在低频的趋势，同时底层成为最差的情况。因此，仿真曲线中注出了缩小了的噪声容限。此外，因为各层的电容效应均相同，故谐振频率基本一致。

总结一下，直到谐振频率为止，底层的 AC 阻抗最差。此点以后顶层具有稍大的阻抗。鉴于热约束声明底层最倾向于包含大电流消耗模块，底层的供电噪声（即电流阻抗积）将成为 3D 芯片实现应重点关注的问题。

上述讨论的目标是为了提供 3D 芯片电源传输的一些定量理解。需要指出的是，这些数据与特定的工艺相关并随工艺改变而变化。例如，如果工艺允许更低电阻和更小面积的 TSV，那么路径中的阻抗瓶颈将出现在供电焊盘上，同时 PSN 模型需要考虑该因素。此外，不考虑该因素时，PSN 仍最有可能成为 3D 设计的关键

问题。

3.3.3 控制 PSN 噪声的设计技术

严重电源传输瓶颈的出现使一种全新的 3D 芯片电源传输方案成为必需。本节中，我们将介绍针对该问题几种可能的解决方案。

3.3.3.1 片上电压调制

一种解决 3D 芯片电源传输问题的方法（也适用于传统 2D 芯片）是让DC-DC 转换模块尽量靠近处理器，如图 3-9[37] 所示。增强外部电压和本地降压转换能保证穿过外部封装的电流 I_{ext} 较小，并降低按比例缩小对外部封装阻抗的要求。此外，该负载点（PoL）调制将来自负载与外部封装和去耦电容的全局谐振噪声有效隔离。

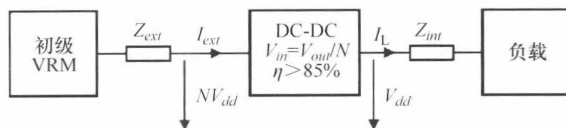


图 3-9　在负载附近 DC-DC 转换器的插入[37] © IEEE 2004

传统上，单片 DC-DC 转换器的效率受片上允许的小物理电感的限制。典型片下 DC-DC 转换需要 1～100μH 的高 Q 电感[17]，而其由于面积需求较大很难在片上实现。随着电源传输问题愈发严峻，研究焦点集中在通过工艺技术构建更紧凑的电感，如薄膜电感[22]，或集中在更高效但更昂贵的通过多相/交错拓扑结构实现的 DC-DC 转换器[42,44]。显然，将这些集成到片上有困难，其需要一整套不同的工艺过程。因此，如 3D 芯片中三维晶圆级堆叠提供的，使堆叠异种工艺的不同晶圆成为可能的方法，是实现片上开关转换器的自然的解决方案。

3.3.3.2 Z 轴方向电源传输

Z 轴或 3D 电源传输[6,16]，其中在 3D 堆叠中 PSN 垂直地与处理器集成，保证了片上 DC-DC 转换具有有吸引力的解决方案。图 3-10 示出了此使用晶圆—晶圆集成的 *z* 轴向电源传输技术的虚拟图[39]。这仍然需要所有的无源器件（包括电感和电容）必须与电源开关和控制电路单片集成。这个想法刺激了相关研究的开展，并促进了此类结构的实现，本章参考文献［39］中使用两个交织的降压变换电路，此开关变换频率为 200MHz 且输出电流为 500mA。在未来，我们将见到这样的多层 3D 芯片，其中整个芯片层用于电压调制，并集成了多种无源器件及其他电路。

z 轴电源传输的一个主要问题是将一个芯片层用于片上 DC-DC 转换器的面积消耗，其尺寸与晶圆—晶圆 3D 工艺中的处理器相当。此外，用于 DC-DC 转换器的高效开关调制器需要大尺寸无源器件的单片实现。另一方面，典型的线性调制

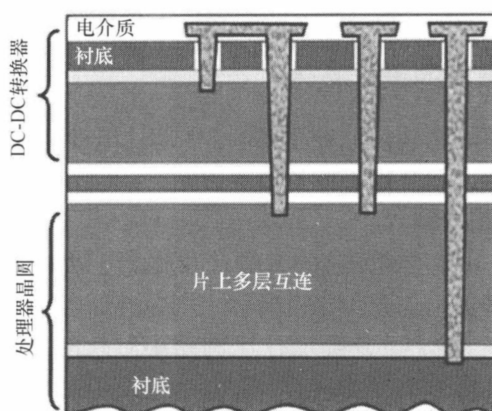


图 3-10　基于 monolithic 电源转换和晶圆—晶圆绑定[39] © IEEE 2007

器，虽然尺寸较小，但转换效率较低。

3.3.3.3　多层电源传输

多层电源传输（MSPD）方案[14,32]是一种实现高效的片上 DC-DC 转换和供电噪声削减的可靠技术的方案。本章参考文献［20］中说明了这种思路对包含了堆叠处理器和存储器的 3D 芯片结构特别有吸引力。

图 3-11 说明了 MSPD 的基本概念。传统供电网络的一种电路图如图 3-11a 所示，其中所有电路从单独的电源汲取电流。图 3-11b 示出了该多层电源传输网络，子电路在两个供电楼层之间工作。“楼层”的概念仅为了准确说明该电源传输方案的本质，与电路物理上堆叠在层中的 3D 芯片结构相反。在此方案中，消耗的电流在“$2V_{dd}$—V_{dd}楼层”，随后在“$2V_{dd}$—Gnd 楼层”循环。由于此内部循环，在总功

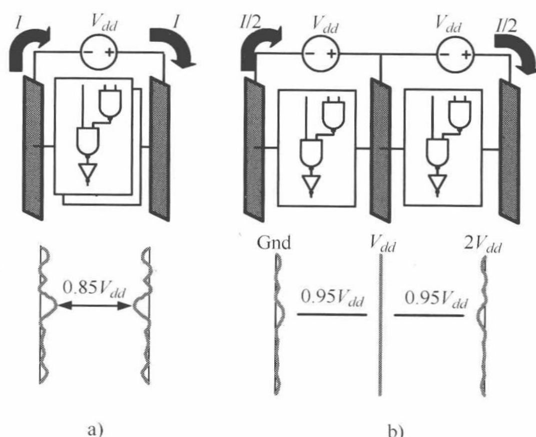


图 3-11　a）传统的电源传输；b）多层电源传输方案[14]。© IEEE 2005

耗基本相同的情况下仅需要相对于传统方案一半的电流。减小的电流对减小供电噪声有利。因此，在最好的情况下，如果两个子电路的电流完全平衡，中间供电路径将吸收零电流。这将使该电源轨的噪声最小化，如图3-11所示。

该技术的主要问题是需要独立的体岛。这在典型体工艺中很难实现。然而，如果我们考虑3D芯片，这些层一开始就是电气隔离的，这让MSPD变得非常有吸引力。

图3-12是简单的3D芯片存储器（M）、存储器（M）、处理器（P）堆叠配置模型。为在模型中描述M和P模块的区别，我们假定后者吸收前者两倍的电流。我们用I和$2I$分别表示这两个电流。TSV带来的层—层路径阻抗用r表示。要注意r是与路径上并行TSV数目成反比的。

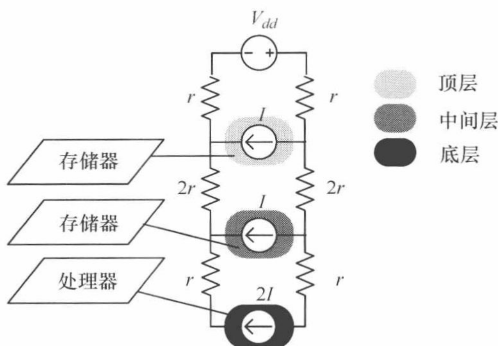


图3-12　3D IC中用于电源传输网络的基准模型。其中处理器假定消耗存储器两倍的电流[20] © IEEE 2008

考虑图3-12中3D芯片的基准电路模型，MSPD应用会引发大量不同的电气等效结构，如图3-13所示。其中，每层—层路径阻抗用R表示。注意到MSPD需要另外一条供电轨，意味着供电轨道的数目增长为原先的3/2。如果我们假定图3-12和图3-13中的所有结构归一化到供电路径通孔的固定数目上，其后的每条供电路径将含2/3的专用通孔。这将成比例对应为3/2倍的阻抗R增长，即$R=1.5r$。

一个细化到3D IC每层的MSPD应用可得到如图3-13a的平衡的PSN配置。其中，每层的电源供电域被分为相等的两亚层，电流在不同层中和之间，从一亚层循环至另一亚层。这种结构平衡了固有的平衡拓扑结构以得到供电噪声级别最大程度的降低。然而，实现它有明显的挑战，特别是在体工艺下体电压隔离的情况下。图3-13b是3D IC中已分离层中实现MSPD应用的粗略结构图。其中，除了保持电流从一个高的电压层循环到较低电压层，每层只有一个专用亚层和单独的体结构。要注意到工作电流在底层的处理器和另外两层的存储器之间循环。这种结构保证了实现其的过程较容易，但由于在处理器和存储器之间有较大的开关电流差异而消除平衡效果，对于降低供电噪声这种结构可能并不有效。在此情况下使MSPD有效需

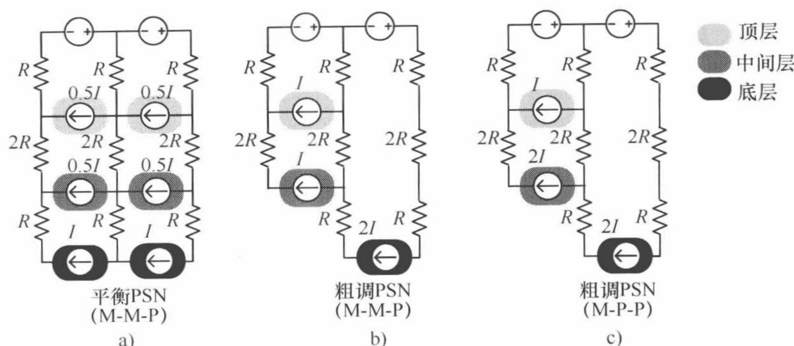


图 3-13 MSPD 在 3D IC 中的应用[20] a）在存储器—存储器—处理器中的平衡 PSN；b）在存储器—存储器—处理器中的粗糙 PSN；c）具有相同层的粗糙 PSN。其中，M 和 P 分别代表存储器和处理器模块，TSV 数目固定为 $R=1.5r$ © IEEE 2008

要将穿过不同供电路径的 TSV 重新排布。相反，如图 3-13c 所示，如果某 3D IC 具有相类似的层，如双处理器堆栈，将在两个处理器模块之间实现开关电流的平衡。这可能提供一种粗化的多层 PSN 思路更有效的实现方法并能在更容易实现的同时潜在地降低 DC 供电噪声。然而，由于中间层与底部热沉隔离，热问题很可能是这里的重要关注点。

表 3-1 给出了以上讨论的总结，说明了如果实现问题可以得到令人满意的重视并可成功，那么 MSPD 可以保证较大的 PSN 噪声削减。从最后一列我们可以看到由于电流循环，PSN 自身的功耗同时降低。这也是 MSPD 的附加优点。

表 3-1 各种 MSPD 方案概述

方　案	架构（M＝内存，P＝处理器）	实施问题	降　噪	功耗降低
平衡 PSN	M-M-P	SOI 中容易，而在体硅中难	最好	最好
粗度 PSN	M-M-P	所需 TSV/电源压焊点的再分配	好	好
粗度 PSN	M-P-P	在体硅和 SOI 中容易，而中间层出现热问题	较好	较好

图 3-14 给出了 MSPD 的全部优点以及其与多种讨论过的方案的比较，其中包括不同漏电流值下的 DC 噪声以及得到的 PSN 功耗缩减量[20]。这些值以对应黑体表示的所谓非 MSPD 的 3D 集成电路模型为准归一化。很明显，MSPD 技术基于不同的拓扑结构和漏电流量，保证了 20% ~40% 的 DC 噪声削减。我们有趣地发现在 MSPD 情况下 DC 噪声随漏电流削减。

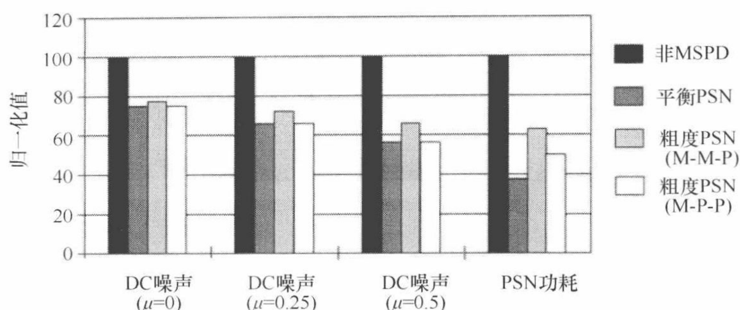


图 3-14　用于不同机制的 DC 噪声和 PSN 电源[20]。
其中，μ 代表漏电流与总工作电流的比值© IEEE 2008

3.3.4　控制 PSN 噪声的 CAD 技术

3.3.4.1　去耦电容定位

现有几种增加电源网格可靠性并控制电源网格噪声的技术是可行的，如线网加宽、网格拓扑结构优化，以及去耦合电容的插入[36]。所有这些技术中，去耦电容是可证明的减小瞬态噪声最有效的方法。去耦电容作为本地电流池，能满足由功能模块/单元的电流需要引起的突然的电涌，并且同时保持供电电压级别相对稳定。对于谐振噪声使用去耦电容的有源/无源阻尼方法也被提出[12,13,46]。

传统实现去耦电容的工艺是基于广泛用在鲁棒电源传输网络设计中的 SiO_2 基结构的。三维电源网格优化在本章参考文献［19，27，43］中有所研究。不同于 2D 情况，在 3D 电源网格优化中使用 CMOS 去耦电容出现了新的考虑因素。具体如下：

1）由于 CMOS 去耦电容通常使用器件层之上的空白区域制造，它们必须与 TSV 区域或者 3D 通孔的接触焊盘争夺空白空间，这导致新的资源分配问题。解决该分配问题的一种方法是增加芯片尺寸以给 CMOS 去耦电容留出空间。然而，3D 电路相对 2D 电路的一个优点正是它们减小了芯片的尺寸：增加芯片尺寸与此优点背道而驰。

2）漏电流功耗是 3D 电路设计的一个重要问题。添加到 3D 电路上的 CMOS 去耦电容将消耗额外的漏电流功耗，并使情况恶化。在新型高 k 电介质提出的情况下，它们尚未广泛使用并且即使它们被采用，也仅能临时性地减轻栅电极漏电问题。

本章参考文献［49］中的研究提出了在 3D 电源网格中的去耦电容定位方法，同时使用了传统 CMOS 去耦电容和金属—绝缘层—金属（MIM）去耦电容。与 CMOS 电容在器件层中构建不同，MIM 电容在金属层间制造。这些结构具有高电容密度和低漏电流密度[3,18,34,35,40,50]。

图 3-15 示出了 3D 电路中 CMOS 和 MIM 电容的位置。MIM 去耦电容在每个 2D 层中通常使用顶部两层金属制造。MIM 去耦电容的一个显著优点源于其极低的漏电流。在本章参考文献［34］中记录的 250nF 的 MIM 去耦电容漏电流约为 1.0 ×

10^{-8}A（其中漏电流密度为 3.2×10^{-8}A/cm^2），与此 MIM 并联的 25nF CMOS 去耦电容漏电流约为 3.2×10^{-6}A（其中漏电流密度为 1.45×10^{-4}A/cm^2）。

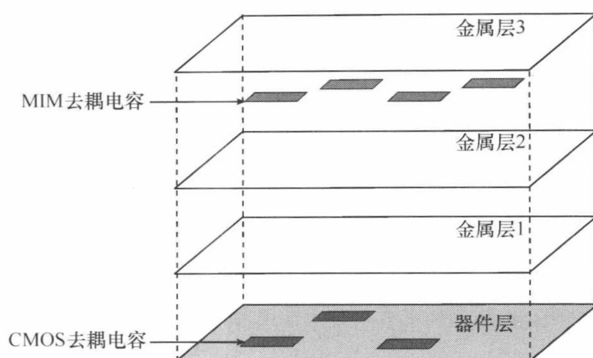


图 3-15 在一个 3 层金属的 2D 芯片层中的 MIM 和 CMOS 去耦电容[49] © IEEE 2009

然而，MIM 去耦电容不能无条件地用于取代 CMOS 去耦电容，因为它们的使用有代价，即引入了对试图穿过它们线网的布线禁区。在本章参考文献［49］中，同时使用 CMOS 和 MIM 去耦电容的规划问题归结为线性编程（LP）问题，并且提出了一种高效拥塞敏感算法来优化电源供电噪声，试图找到布线拥塞恶化和漏电流功耗增加的平衡。

一种迭代流程被用来解决去耦电容定位问题。由于两个原因，在每次迭代中仅有相对少量的去耦电容定位到当前电路。第一，去耦电容定位问题是高阶非线性的，并且该迭代过程允许受控于解决线性程序序列的优化过程，每次迭代一回。为了使能这些线性程序的公式，必须使用去耦电容值是线性的模型为噪声混叠和拥塞建模。第二个原因与所做简化有关，该简化避免了可能造成拥塞和噪声混叠的简化线性模型失效过多去耦电容的分布；这些模型在微扰条件下是可预测的。

试验结果显示，单独使用 CMOS 去耦电容不足以克服混叠；MIM 去耦电容的使用会导致高阶拥塞。将两种去耦电容的最优混合在低漏电流功耗的基础上同时满足了拥塞和噪声约束。

3.3.4.2 用于 3D IC 的自动 MSPD 指定

3.3.3.3 节中介绍的 MSPD 可以在解决优化问题的基础上自动实现[47,48]，其将两层问题视为两层间模块指定问题之一。MSPD 电路设计中一个重要的考虑是局部地保持工作在不同 V_{dd} 域中的逻辑块的电流平衡，因为反之不平衡电流将流过稳压器并浪费掉。另一个需要考虑的重要问题是在哪个设计层次电路应该被分块到不同的电压域。注意到如果逻辑块用于驱动另一个工作在不同 V_{dd} 域中的逻辑电路，其输出端需要加电平转换器。电平转换器占用芯片面积并引发电路的额外延时。

模块指定问题在布局规划层面上是重点，前提是模块数目通常不太大，并且它们的面积很大程度上被忽略了。假定 K 个稳压器分布在芯片上：这些稳压器设计出色并输出稳定电压 V_{dd}。

每个稳压器由嵌入 V_{dd} 网格的点代表。如图 3-16a 所示，芯片被相应地分为 K 个区域，因此每个区域中有一个稳压器并且第 i 个区域包含片上所有主要从第 i 个稳压器得到或输出电流的点。将芯片分割成这些不重叠区域的工作可通过使用细网格来网格化芯片区域实现，其中每个模块的网格单元属于或者说每个单元可称为属于由最近的电压稳压器控制的区域。

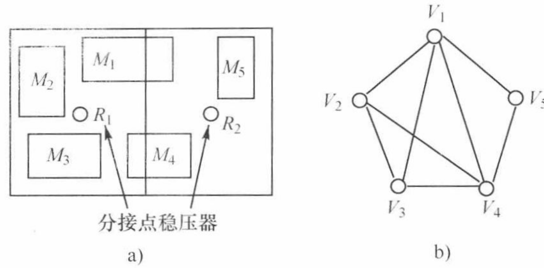


图 3-16　图的构建：a）把芯片分块得到每个均由一个稳压器控制的互不相连的区域。b）构建图，其中节点 V_i 对应模块 M_i[48] © IEEE 2007

一旦芯片划分为不相连的区域，就假定任意“非平衡”电流，即未循环至下一层特定区域的电流，通过了同一区域的稳压器并浪费掉了。如果模块位于多个区域之间的边界，它将被分解为几个子模块，每个区域有一个与其重叠的子模块，并且约束所有的子模块必须指定到相同的电压域。

让我们关注对应特定电压稳压器的一个特定区域。假定位于此区域的模块为 M_1，$M_2 \cdots M_n$，其中流过模块 M_i 的电流是时间 t 的函数，由 $I_i(t)$ 给出。因为电压稳压器仅能对应低到中频分量的非平衡电流，而高频分量通常由片上去耦电容处理，于是我们预处理通过准确到周期的功耗模拟得到的电流轨迹，从而去除电流信号中的高频分量。因此，$I_i(t)$ 应被理解为流经模块 M_i 的仅包含低频到中频分量的电流。

如果将 0/1 整型变量 x 与模块 M_i 联合定义为

$$x_i = \begin{cases} 0 & M_i\text{在 } 2V_{dd} \text{ 和 } V_{dd} \text{ 之间操作} \\ 1 & M_i\text{在 } V_{dd} \text{ 和 } G_{nd} \text{ 之间操作} \end{cases} \tag{3-22}$$

那么 t 时刻流经电压稳压器的总电流约为

$$I_R(t) = \left| \sum_{i=1}^{n} I_i(t) \cdot (1 - x_i) - \sum_{i=1}^{n} I_i(t) \cdot x_i \right| = \left| \sum_{i=1}^{n} I_i(t) \cdot (1 - 2x_i) \right| \tag{3-23}$$

该问题可视为映射为一种最大化各分块间切割区域面积的图分块，其中边的权重由下式给出：

$$w(V_i, V_j) = \left[\sum_{k=1}^{K} \frac{S_{ik} S_{jk}}{S_i S_j} \right] \overline{I_i(t) I_j(t)} \tag{3-24}$$

式中，S_i 代表第 i 个模块的面积；第 i 个模块与第 k 个模块重叠的区域由 S_{ik} 表示。

式（3-24）直观上意义在于对于任意模块对，仅在芯片上位于相同区域的部分按它们之间的相关系数计算。如果模块 M_i 和 M_j 彻底分成了两个不相连的区域，

权重 $w(V_i, V_j)$ 将为零，并因此可去除图中对应边。其结构示例见图 3-16b。一种 Fidducia-Mattheyses-式方法[8]用来快速找到最大化切割数的分块方式。

本章参考文献［47］中的试验结果显示所述方法在构建基于 SOI 工艺的 2D 和 3D 芯片的多层电源网格分块均有效，其中来自多层的块可以在一个芯片层共存。结果显示基于分块的方法在通过系统回收大量能量时是成功的，同时基于分块的结果的质量优于退火方法。

3.4 结论

本章中，我们广泛地分析了未来 3D 芯片中的热和电源的传输问题。两个问题有一个共同来源，因为它们都是 3D 集成电路中单位面积增长的电流引起的，并由此引发了严重的可靠性问题和潜在的逻辑错误。热问题将在本书中进行更深入的分析，但我们已经通过设计和 CAD 方法提供了克服电源传输问题的整体解决方案。

参考文献

1. HotSpot. available at http://lava.cs.virginia.edu/HotSpot/index.htm.
2. M. A. Alam, B. E. Weir, and P. J. Silverman. A study of soft and hard breakdown - part II: Principles of area, thickness, and voltage scaling. *IEEE Transactions on Electron Devices*, 49(2):239–246, February 2002.
3. M. Armacost, A. Augustin, P. Felsner, Y. Feng, G. Friese, J. Heidenreich, G. Hueckel, O. Prigge, and K. Stein. A high reliability metal insulator metal capacitor for 0.18 μm copper technology. In *Proceedings of the IEEE International Electronic Devices Meeting*, pp. 157–160, 2000.
4. Semiconductor Industry Association. International technology roadmap for semiconductors (ITRS), 2007. http://public.itrs.net/.
5. J. A. Burns, B. F. Aull, C. K. Chen, C. L. Keast, J. M. Knecht, V. Suntharalingam, K. Warner, P. W. Wyatt, and D. Yost. A wafer-scale 3-D circuit integration technology. *IEEE Transactions on Electron Devices*, 53(10):2507–2516, October 2006.
6. S. Chandrasekaran, J. Sun, and V. Mehrotra. Vertically packaged switched-mode power converter, 2006. US Patent #7012414.
7. L. O. Chua and P.-M. Lin. *Computed-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Prentice-Hall, Englewood Cliffs, NJ, 1975.
8. C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 175–181, 1982.
9. G. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD, 3rd edition, 1996.
10. B. Goplen. *Advanced Placement Techniques for Future VLSI Circuits*. PhD thesis, University of Minnesota, Minneapolis, MN, 2006.
11. B. Goplen and S. S. Sapatnekar. Efficient thermal placement of standard cells in 3D ICs using a force directed approach. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 86–89, 2003.
12. J. Gu, H. Eom, and C. H. Kim. Sleep transistor sizing and control for resonant supply noise damping. In *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 80–85, 2007.

13. J. Gu, H. Eom, and C. H. Kim. A switched decoupling capacitor circuit for on-chip supply resonance damping. In *Proceedings of the IEEE International Symposium on VLSI Circuits*, pp. 126–127, 2007.
14. J. Gu and C. H. Kim. Multi-story power delivery for supply noise reduction and low voltage operation. In *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 192–197, 2005.
15. E. Hailu, D. Boerstler, K. Miki, J. Qi, M. Wang, and M. Riley. A circuit for reducing large transient current effects on processor power grids. In *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 2238–2245, 2006.
16. J. A. Harrison and E. R. Stanford. Z-axis processor power delivery system, 2003. US Patent #6523253.
17. P. Hazucha, G. Schrom, J. Hahn, B. A. Bloechel, P. Hack, G. E. Dermer, S. Narendra, D. Gardner, T. Karnik, V. De, and S. Borkar. A 233-MHz 80%–87% efficient four-phase DC-DC converter utilizing air-core inductors on package. *IEEE Journal of Solid-State Circuits*, 40(4):838–845, April 2005.
18. H. Hu, S.-J. Ding, H. F. Lim, C. Zhu, M. F. Li, S. J. Kim, X. F. Yu, J. H. Chen, Y. F. Yong, B. J. Cho, D.S.H. Chan, S. C. Rustagi, M. B. Yu, C. H. Tung, A. Du, D. My, P. D. Foot, A. Chin, and D.-L. Kwong. High performance ALD HfO_2-Al_2O_3 laminate MIM capacitors for RF and mixed signal IC applications. In *Proceedings of the IEEE International Electronic Devices Meeting*, pp. 15.6.1–15.6.4, 2003.
19. G. Huang, M. Bakir, A. Naeemi, H. Chen, and J. D. Meindl. Power delivery for 3D chip stacks: Physical modeling and design implication. In *Proceedings of the IEEE Electrical Performance of Electronic Packaging Meeting*, pp. 205–208, 2007.
20. P. Jain, T. Kim, J. Keane, and C. H. Kim. A multi-story power delivery technique for 3D integrated circuits. In *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 57–62, 2008.
21. C. Keast, B. Aull, J. Burns, N. Checka, C.-L. Chen, C. Chen, M. Fritze, J. Kedzierski, J. Knecht, B. Tyrrell, K. Warner, B. Wheeler, D. Shaver, V. Suntharlingam, and D. Yost. 3D integration for integrated circuits and advanced focal planes, 2007. Available at http://vmsstreamer1.fnal.gov/VMS_Site_03/Lectures/Colloquium/070228Keast/index.htm .
22. K. H. Kim, J. Kim, H. J. Kim, S. H. Han, and H. J. Kim. A megahertz switching DC/DC converter using FeBN thin film inductor. *IEEE Transactions on Magnetics*, 38(5):3162–3164, September 2002.
23. P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra. Efficient full-chip thermal modeling and analysis. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 319–326, 2004.
24. P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra. IC thermal simulation and modeling via efficient multigrid-based approaches. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(9):1763–1776, September 2006.
25. D. L. Logan. *A First Course in the Finite Element Method*. Brooks/Cole Publishing Company, Pacific Grove, CA, 3rd edition, 2002.
26. R. Mahajan, R. Nair, V. Wakharkar, J. Swan, J. Tang, and G. Vandentop. Emerging directions for packaging technologies. *Intel Technology Journal*, 6(2):62–75, May 2002.
27. J. R. Minz, S. K. Lim, and C.-K. Koh. 3D module placement for congestion and power noise reduction. In *Proceedings of the Great Lakes Symposium on VLSI*, pp. 458–461, 2005.
28. N. Na, T. Budell, C. Chiu, E. Tremble, and I. Wernple. The effects of on-chip and package decoupling capacitors and an efficient ASIC decoupling methodology. In *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 556–567, 2004.
29. M. N. Özişik. *Heat Transfer: A Basic Approach*. McGraw-Hill, New York, NY, 1985.
30. H. Qian, S. R. Nassif, and S. S. Sapatnekar. Random walks in a supply network. In *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 93–98, 2003.
31. H. Qian and S. S. Sapatnekar. Hierarchical random-walk algorithms for power grid analysis. In *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 499–504, 2004.

32. S. Rajapandian, K. Shepard, P. Hazucha, and T. Karnik. High-tension power delivery: Operating 0.18 μm CMOS digital logic at 5.4 V. *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 298–599, February 2005.
33. V. Reddy, A. T. Krishnan, A. Marshall, J. Rodriguez, S. Natarajan, T. Rost, and S. Krishnan. Impact of negative bias temperature instability on digital circuit reliability. In *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 248–254, 2002.
34. D. Roberts, W. Johnstone, H. Sanchez, O. Mandhana, D. Spilo, J. Hayden, E. Travis, B. Melnick, M. Celik, B. W. Min, J. Edgerton, M. Raymond, E. Luckowski, C. Happ, A. Martinez, B. Wilson, P. Leung, T. Garnett, D. Goedeke, T. Remmel, K. Ramakrishna, and B.E. Jr. White. Application of on-chip MIM decoupling capacitor for 90 nm SOI microprocessor. In *Proceedings of the IEEE International Electronic Devices Meeting*, pp. 72–75, 2005.
35. H. Sanchez, B. Johnstone, D. Roberts, O. Mandhana, B. Melnick, M.Celik, M. Baker, J. Hayden, B. Min, J. Edgerton, and B. White. Increasing microprocessor speed by massive application of on-die high-k MIM decoupling capacitors. In *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 2190–2199, 2006.
36. S. S. Sapatnekar and H. Su. Analysis and optimization of power grids. *IEEE Design & Test*, 20(3):7–15, 2003.
37. G. Schrom, P. Hazucha, J. Hahn, V. Kursun, D. Gardner, S. Narendra, T. Karnik, and V. De. Feasibility of monolithic and 3D-stacked DC-DC converters for microprocessors in 90*nm* technology generation. In *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 263–268, 2004.
38. K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayan, and D. Tarjan. Temperature-aware microarchitecture. In *Proceedings of the ACM International Symposium on Computer Architecture*, pp. 2–13, 2003.
39. J. Sun, J. Lu, D. Giuliano, T. P. Chow, and R. J. Gutmann. 3D power delivery for microprocessors and high-performance ASICs. In *Proceedings of IEEE Applied Power Electronics Conference*, pp. 127–133, 2007.
40. Y. L. Tu, H. L. Lin, L. L. Chao, D. Wu, C. S. Tsai, C. Wang, C. F. Huang, C. H. Lin, and J. Sun. Characterization and comparison of high-k metal-insulator-metal (MiM) capacitors in 0.13 μm cu BEOL for mixed-mode and RF applications. In *Proceedings of the IEEE International Symposium on VLSI Circuits*, pp. 79–80, 2003.
41. A. Waizman. CPU power supply impedance profile measurement using FFT and clock gating. In *Proceedings of the IEEE Electrical Performance of Electronic Packaging Meeting*, pp. 29–32, 2003.
42. J. Wibben and R. Harjani. A high efficiency DC-DC converter using 2 nH on-chip inductors. In *Proceedings of the IEEE International Symposium on VLSI Circuits*, pp. 22–23, 2007.
43. E. Wong, J. Minz, and S. K. Lim. Decoupling capacitor planning and sizing for noise and leakage reduction. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 395–400, 2006.
44. P. Wong, P. Xu, P. Yang, and F. C. Lee. Performance improvements of interleaving VRMs with coupling inductors. *IEEE Transactions on Power Electronics*, 16(4):499–507, July 2001.
45. J. H. Wu. *Through-Substrate Interconnects for 3-D Integration and RF Systems*. PhD thesis, Department. of EECS, Massachusetts Institute of Technology, October 2006.
46. J. Xu, P. Hazucha, M. Huang, P. Aseron, F. Paillet, G. Schrom, J. Tschanz, and C. Zhao. On-die supply-resonance suppression using band-limited active damping. In *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 286–603, 2007.
47. Y. Zhan and S. S. Sapatnekar. Automated module assignment in stacked-Vdd designs for high-efficiency power delivery. *ACM Journal on Emerging Technologies in Computing Systems*, 4(4):1–20, 2008.
48. Y. Zhan, T. Zhang, and S. S. Sapatnekar. Module assignment for pin-limited designs under the stacked-Vdd paradigm. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 656–659, 2007.

49. P. Zhou, K. Sridharan, and S. S. Sapatnekar. Congestion-aware power grid optimization for 3D circuits using MIM and CMOS decoupling capacitors. In *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 179–184, 2009.
50. P. Zurcher, P. Alluri, P. Chu, A. Duvallet, C. Happ, R. Henderson, J. Mendonca, M. Kim, M. Petras, M. Raymond, T. Remmel, D. Roberts, B. Steimle, J. Stipanuk, S. Straub, T. Sparks, M. Tarabbia, H. Thibieroz, and M. Miller. Integration of thin film MIM capacitors and resistors into copper metallization based RF-CMOS and Bi-CMOS technologies. In *Proceedings of the IEEE International Electronic Devices Meeting*, pp. 153–156, 2000.

第 4 章　热敏感 3D 布局规划

Jason Cong　Yuchun Ma

三维集成使布局规划更加困难，原因在于多器件层显著地扩大了解空间，并且功率密度的增加加剧了热问题。本章介绍了带有 2D 块和 3D 块的 3D 布局规划的算法。作为简要介绍的基于多种表达式的随机优化的补充，本章介绍了分析式的方法。多种关于线网长度、面积和温度的 3D 布局规划技术影响的说明通过实验结果给出。

4.1　介绍

3D 芯片设计提供了逻辑块拓扑排布的另一个维度。因此，在采用 3D 技术时物理设计工具占据重要的地位。作为物理设计流程中的关键步骤，布局规划严重影响着最终设计的性能。因为多个芯片层显著扩大了解空间，同时功率密度的增加导致了热问题，所以三维集成使布局规划问题变得更加困难。因此，进入 3D 设计阶段极大地增加了布局规划问题的复杂度。

1）3D 集成电路布局规划的设计空间随有效层数的增加按指数级增加。

2）附加的温度约束或温度最小化目标使优化过程复杂化。需要在面积、线网长度和热特征参数间折中来完成优化。同时由于 3D 芯片中的高温，解决封闭的温度/漏电功耗反馈环以准确估计或优化两者是必要的。

3）多芯片层堆叠实现了块间迟滞时间的缩减。它还可用于在多芯片层实现块中，以减小块内线网迟滞时间。多层块的使用需要新的基本物理设计方法以应对三维设计空间。

因此，开发考虑 3D 设计约束的热敏感布局规划工具势在必行。3D 布局规划的目标是在不违反某些设计约束的情况下优化某些目标参数，从而在多芯片层不重叠地打包块。根据块的表示方式，我们可将 3D 布局规划问题分为两类。第一类是 2D 块的 3D 布局规划，其中每个块是 2D 的长方形并且每层的打包可视为 2D 布局规划。2D 块的 3D 布局规划可表示为 2D 表示的矩阵（2D 矩阵），其中每个代表所有位于一个器件层的块。第二类的 3D 布局规划包含 3D 块，其中每个块视为在 Z 轴上具有非零高度的立方块。在该情况下，现有的 2D 表示不再适用，我们需要新的表示方法。

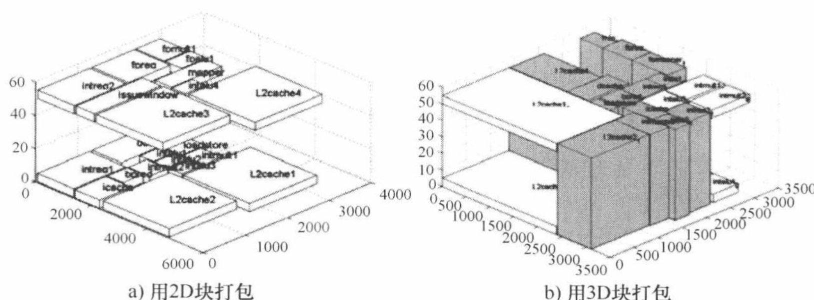
本章讨论了这两种方法及其在 3D 集成电路中的应用。4.2 节阐述了 3D 布局规划问题；4.3 节和 4.4 节分别介绍了用于含 2D 和 3D 块的 3D 布局规划的优化的表

示方法。在 4.5 节中，介绍了 3D 布局规划的优化技术。在通常使用的模拟退火优化方法以外，还介绍了解析方法，例如力导向法。4.6 节提供了多种技术的实验结果。最后，我们在 4.7 节给出了结论。

4.2　问题说明

与传统 2D 布局规划相似，3D 布局规划的目标也是打包面积小、线网长度短、功耗低和性能高。如第 3 章中说明的，虽然 3D 集成有很多潜在的优势，但热分布成为贯穿 3D 设计每个阶段的关键问题。因此，3D 布局规划将块互不重叠地分布在一定数目的芯片层上从而使如芯片面积、线网长度、层间过孔数据和最大片上温度的设计指标优化或达到某些设计约束。

随着 z 轴的加入，不仅 2D 块可在多芯片层间扩展，而且某些独立器件可设计折叠为多层块形式以减小块内线网迟滞和功耗。近期的研究提出了多种 3D 构架结构，包括 3D 缓存[30,9,28]，3D 寄存器堆[31]、3D 运算单元[25]和 3D 指令调度器[26]。具有不同层数的 3D 部件可视为需打包到 3D 空间中的立方块。z 方向的尺寸代表层信息。因此，在 3D 布局规划中，需要打包的块可为 2D 块或 3D 块。图 4-1a 示出了 Alpha 21264 的两层打包，其中所有的块为 2D 块，而图 4-1b 示出了一些 3D 块的打包。每个 3D 部件的实现因不同的面积—延时—功耗的折中有多种可能的选择。如图 4-1b 所示，最优的布局规划可能是占据单个器件层的结构单元的子集，而其他的可实现在多层上，潜在地具有 z 方向上的不同高度。根据块的表示法，可将 3D 布局规划问题分为两类，即仅含 2D 块的 3D 布局规划和可能含 3D 块的 3D 布局规划。



a) 用2D块打包　　b) 用3D块打包

图 4-1　三维布局规划

4.2.1　含二维块的三维布局规划

虽然含 2D 块的 3D 打包可视为多层堆叠的 2D 打包，但在芯片层级上存在附加问题，即当大量的有源器件打包到相当小的区域中时，会导致功率密度远高于其对

应的 2D 电路。结果是除了一般的面积、线网长度的打包指标之外，热问题成为设计指标中的首要问题。此处，我们可以将含 2D 块的 3D 布局规划进行一下说明。

有一个含 2D 块的 3D 布局规划问题的实例，该实例块由一系列块 $\{m_1, m_2, \cdots, m_n\}$ 组成。块 m_i 是 $W_i \times H_i$ 面积为 A_i 的长方形，高宽比为 H_i/W_i，功率密度为 PD_i。每个块均可自由旋转。芯片层数目固定为 L。令三元向量（x_i，y_i，l_i）代表块 m_i 的底部左侧角顶点的坐标，其中 $1 \leqslant l_i \leqslant L$。3D 布局规划 F 即为每个块 m_i 指定坐标（x_i，y_i，l_i）并使其互不重叠。3D 布局规划算法通常的目标是最小化下列各项：①芯片峰值温度 $T_{\max}$；②总线网长度（或总功耗）；③芯片面积。芯片面积是所有层最大高度和宽度的乘积。线网长度是半周长线网长度估计。此外，有些另外的设计目标，例如噪声、性能和芯片层间过孔的数目等需要被同时考虑。同时，一些设计约束需要包括在内，例如预打包块（受约束的块的位置是预定义的），对齐约束（某些特定块约束为沿 x、y 或 z 方向对齐）。

因为含 2D 块的 3D 布局规划可表示为由 2D 表示的矩阵，所以 2D 布局规划算法可以通过在优化技术中引入新的操作扩展到多层设计。虽然 2D 设计的布局规划是一个充分研究过的问题，但加入芯片层的信息后，3D IC 布局规划的设计空间以指数级增加。Li 等人说明了给定一个含 n 个块的布局规划问题，L 层 3D 布局规划的解空间相对于 2D 的情况增加了 $n^{L-1}/(L-1)!$ 倍[11]。虽然多层设计可以表示为 2D 打包的矩阵，但是特定优化技术仍需要充分研究。热敏感优化在 3D 设计中极为关键。

4.2.2 含三维块的三维布局规划

细颗粒三维集成提供了更小的块内线网延时并同时优化了功耗（更多的细节在附录中提供）。每个组件的实现因多种不同的配置而有多种选择。因此，组件可通过不同的堆叠技术实现在多层之上，如四层或两层缓存。但局部看来最优的独立单元实现不一定是对于整个多层芯片最优的设计。为得到多个指标的折中，在打包设计中实现具有不同 z 轴向高度的立方块是可能的。因此，应开发立方打包的算法。此算法可以在最小体积内互不交叠的排布那些给定的以立方体形式存在的电路组件。在每个关键部件具有多种实现方式的情况下，块的实现是部分定义的。没有物理信息就不可能得到用于最终芯片的最优组件实现。于是，含 3D 块的 3D 布局规划不能仅要确定块的坐标，还要能选择组件的配置，如层数目和分块方式。因此，我们可以将含 3D 块的 3D 打包描述如下：

给定 3D 块列表：假设对于块 i，有 k 个不同的实现方式，记录在备选表中 $\{c_1^i, c_2^i, \cdots c_k^i\}$。每个备选 c_j^i 宽（w_j^i），高（h_j^i），层数（z_j^i），延时（d_j^i），功耗（p_j^i）（假定每层具有相同的功耗）。目标是生成一个优化了芯片面积、最大片上温度等指标的布局规划。同时，层的数量通常固定，意味着给定的层数约束为 Z_{con}，且块不能超过该层数约束。在本章中，我们将提供可以表示 4.4 节中立方打包的几

种典型的表示方式。

4.3　含二维块的三维布局规划表示法

因为含 2D 块的 3D 布局规划可表示为 2D 表示的矩阵，所以 2D 布局规划算法可通过在优化技术中引入新操作以扩展到可处理多层设计。在讨论详细的 3D 操作前，先简要地介绍基本的 2D 表示法。这是 3D 布局规划优化的基础。

4.3.1　二维表示法的基本表示

块间的几何关系通常通过布局规划区域的矩形分割定义。为了约束解空间尺寸，3 种不同的分割方法被提出。其对应的布局规划结构称为切片[34,23]、镶嵌[15,7,42]和总体布局[20,18]。切片布局规划是镶嵌布局规划的特例，镶嵌布局规划是总体布局规划的特例。切片、镶嵌和总体布局规划的解空间关系如图 4-2 所示。

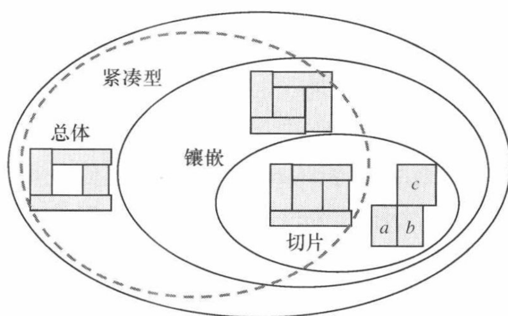


图 4-2　切片、镶嵌和总体布局规划的解空间的关系

切片布局规划可通过使用水平线或垂直线递归切割矩形得到，每次切割将矩形分为两部分。在镶嵌布局规划中，布局规划区域分割为精确的空间，使得每个空间均由一个且仅一个块占据。总体布局规划类似镶嵌布局规划，因其允许非切片结构。然而，布局规划区域可分割为多个空间，其数目大于块数，这样有一些空间不被任何块占据。除这三类以外，还有一种紧凑打包。当其他块固定时如果没有块可左移（下移），确立的打包称为 L- compat（B- compat）。如图 4-2 所示，3 个块 a、b 和 c 的打包为切片和镶嵌，但不是紧凑打包。因此，紧凑打包是一种特殊的打包类型，其解空间与切片和镶嵌的解空间交叠，但不完全包含它们。

由于 2D 和 3D 矩形打包问题是 NP 困难问题，大多数布局规划算法基于如模拟退火的随机组合优化技术。在优化过程中，通常使用拓扑表示，因为它们保证了所有编码的打包不重叠，且每个拓扑表示一种易于移动的方式包含了块间的相对位置信息（如之右，之左，之上，之下）。切片结构可译为切片树[22]或波兰表达式[34]。在序列对[18]作为第一个总体布局规划的表示法提出后，多种（如 BSG[19]，

O 树[6]，B*树[1]，CBL[7]和 TCG[13,14]）布局规划表示法被提出。这些表示法为解的探索提供了不同程度的冗余和效率。以下，我们简要的介绍几种典型的表示法。

4.3.1.1 切片结构

因为切片结构能通过垂直或水平线递归分块，所以称为切片树[22,23,34]的定向根二叉树用于表示块间的分块过程（见图 4-3）。树的每个内部节点用“ * ”或“ + ”操作数表示，分别对应垂直和水平切割。每个叶子对应一个基本块并用从 1 到 n 的数表示（n 是块的数目）。

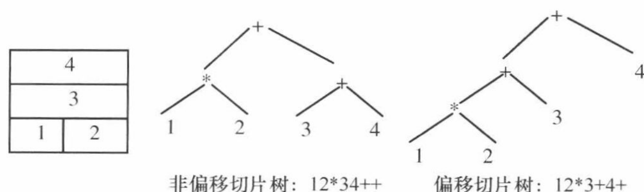


图 4-3 切片布局规划的切片树状表示和波兰（polish）表达式表示

然而，对于给定的切片布局规划，存在超过一个的切片树表示。为了得到所有切片布局规划的非冗余表示，Wong 和 Liu[34]提出了一种特别的切片树，称为偏移切片树（SST）。SST 是没有节点并且其右侧子节点具有相同的标号（见图 4-3）。SST 的逆向遍历，称为归一化波兰表达式（NPE），它们使用它作为布局规划表示法。如果波兰表达式序列中没有连续的“ * ”或“ + ”则称其为归一化。本章参考文献［34］中证明长度为 $2n-1$ 的归一化波兰表达式集和 n 个块的切片布局规划有一一对应关系。打包可通过在线性时间内扫描切片树或归一化波兰表达式实现。

4.3.1.2 镶嵌结构

本章参考文献［7］中介绍了镶嵌布局规划，其有以下特征：

1）布局规划中不存在空闲空间，即每个矩形指定到一个且仅一个块。如图 4-4所示，某些具有空闲空间的结构无法用镶嵌布局规划表示。

2）在非交叉片段滑动以适应块尺寸的之前和之后，其拓扑结构是等价的。

3）不存在两个片段在同一点相遇的退化情况。如果退化情况发生，片段会被分开并向两侧滑开一小段距离（退化情况的进一步讨论见 4.3.2.2 节）。

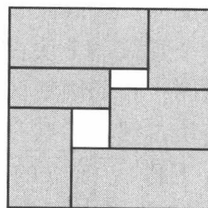
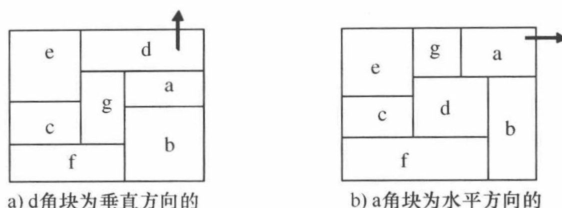


图 4-4 带有空闲空间的打包

角块列表[7,15,40]，双位序列[39]，Q 序列[42]等提出来代表镶嵌布局规划。在以下的讨论中，我们将简要介绍角块列表。

角块列表（CBL）使用（S，L，T）的三元列表，其中 S 记录块 ID 的序列。L 记录每个块的方向：$L_i=0$ 代表块 i 从上方覆盖了其他块；$L_i=1$ 代表块 i 从右侧覆

盖了其他的块（如图 4-5 所示）。位表 T 记录了有多少个块在其打包时被一个块所覆盖。在表 T 中，每个子表的长度，其中子表具有一定数目连续的“1”并以“0”结尾，对应被块 i 覆盖的块数目。图 4-6 是一个非切片布局规划及其对应的 CBL 的例子。在图 4-6 中，块 g 覆盖了两个块 {f，e}。因此，子表 {10} 用来在表 T 中表示块 g。



a) d角块为垂直方向的　　b) a角块为水平方向的

图 4-5　角块的定向

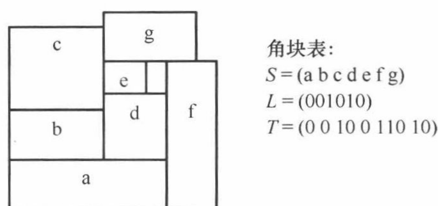


图 4-6　非切片布局规划及其对应的 CBL 表

要注意在一个 CBL 表中，子表 T_i 中连续“1”的个数可能大于在对应方向上存在的未覆盖块。为修正该问题，结束位“0”可自动/有效插入，说明该块将覆盖对应方向上所有存在的块。假定存在块的数目为 t。当扫描块 i 的表 T 时，如果有 t 个连续的“1”，那么“0”可以立刻自动/有效插入在那些连续的“1”之后来结束该子表 T_i。于是表 T 的扫描可以继续进行下一个块。因此，对应到镶嵌布局规划的任意 CBL 是可行的。

4.3.1.3　总体布局规划结构

由于允许非切片结构，总体布局规划与镶嵌布局规划类似。然而，打包区域可分割为多于 n 个空间并且使一些空间不被任何块占据。在 20 世纪 90 年代中期，序列对（SP）[18] 和边缘切割线网格（BSG）[19] 出现之前没有对于总体布局规划的有效的拓扑表示法。序列对是对于总体布局规划的出色的表示法并被广泛使用。之后，TCG/TCG-S[13,14] 也被提出来表示总体结构。在本小节中，我们将简要的介绍序列对和 TCG。

序列对：序列对是一对代表 n 个块表的 n 个元素的序列。两种置换（$\Gamma+$，$\Gamma-$）捕捉了每两块间的几何关系。基本上，序列对用以下的方式规定了每对块间的关系：

$$(<\cdots,a,\cdots,b,\cdots>;<\cdots,a,\cdots,b,\cdots>)\rightarrow a\text{ 在 }b\text{ 左侧}$$

$(<\cdots,a,\cdots,b,\cdots>;<\cdots,b,\cdots,a,\cdots>)\rightarrow$a 在 b 上方

每两个块在垂直和水平方向互相约束，并且仅记录这些约束。因此块的位置尽可能推到了左下，同时满足了在序列对中编码的拓扑关系。图 4-7 是一个序列对的例子。

最初的 $O(n^2)$ 次求值算法，来自本章参考文献［18］，在本章参考文献［32］中得到了有效的改进。本章参考文献［32］中的算法在 $O(n\log(n))$ 的时间内运行，用基于在一对加权的序列中计算最长公共子序列的方法对序列对求值。之后的工作本章参考文献［33］改进了本章参考文献［32］中的算法并在不影响结果块位置的情况下将运行时间减小到 $O(n\log(\log(n)))$。

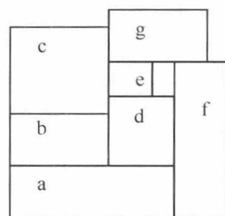


图 4-7　一个打包的序列对：(c b g e d a f, a b c d e f g)

TCG：TCG 描述了基于两个图的块间的几何关系。这两个图称为水平转移闭合图 C_h 和垂直转移闭合图 C_v，其中节点 n_i 代表块 b_i，$C_h(C_v)$ 中的边 (n_i, n_j) 表示块 b_i 在块 b_j 的左侧（下方）。

图 4-8 示出了 5 个块 a、b、c、d、e 的布局及其对应的 TCG 图。在 $C_h(C_v)$ 中节点边的值是对应块的宽度（高度），$C_h(C_v)$ 中的边 (n_i, n_j) 表示 b_i 和 b_j 的水平（垂直）关系。其中，S 和 T 是代表源节点和目标节点的空闲节点。为清晰起见，我们省略图 4-8 中连接空闲结点的转移边。由于 C_h 中存在边（nb，nd），故块 b 在 d 左侧。同理，由于 C_v 中存在边（na，nb），a 在 b 下方。因此，通过穿越约束图找到最长的路径，每个块的位置就确定了。

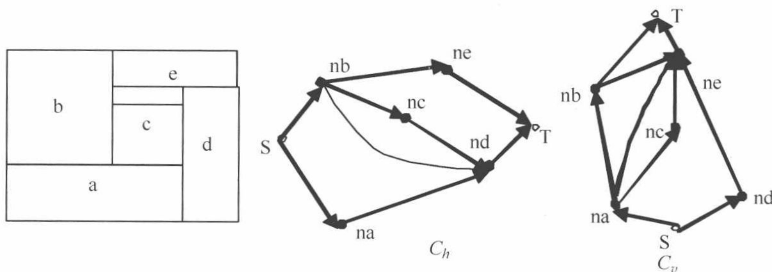


图 4-8　一个打包及其对应的 TCG

4.3.1.4　紧凑结构

总体布局规划表示法的巨大的解空间限制了其在大规模布局规划问题中的应用。O 树[6] 和 B* 树[1] 给出了一种总体布局规划的紧凑版本。与 SP 和 TCG 相比，这两种表示法的解空间更小。然而，它们仅代表部分拓扑信息，并且需要所有块的尺寸以描述精确的布局规划。此外，不是所有可能的矩形分割都可以表示为 O 树和 B* 树。例如，由于块 A 没有推到如图 4-9b 所示的最左侧，图 4-9a 的打包是不

紧凑的。但如果块 A 和 B 之间有许多连接，a 中的打包线网长度将优于 b。因此，某些用序列对或 CBL 表示的打包不能用 O 树和 B^* 树表示。由于 O 树和 B^* 树的结构有些相似，我们以下简要介绍 B^* 树。

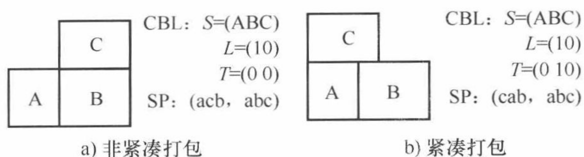


图 4-9　打包示例

B^* 树：O 树和 B^* 树代表一种二叉树的紧凑表示法。其中每个节点对应一个块（见图 4-10）。根节点代表左下的块。例如图 4-10 中的 B3 是左下块。左子节点是其父节点的右侧最下方的邻块，右子节点是其父节点之上的最下方节点并与其父节点 x 轴坐标相同。在图 4-10 中，B5 和 B2 分别是 B3 的左右子节点。给定一个 B^* 树，块位置可通过树的深度优先遍历得到。在块 A 放置到（x_A；y_A）后，我们考虑其子节点 B 并令 $x_B = x_A + w_A$，其中 w_A 是 A 的宽度；y_B 是避免与之前放置的块重叠的最小非负值。在从块 B 的递归返回后，我们考虑 A 的右子节点 C：$x_C = x_B$，同时 y_C 为避免重叠的最小值。该算法能用轮廓数据结构以 $O(n)$ 的时间实现。打包的轮廓定义了其上边沿（参差不齐）并能实现成线段的双链表（见图 4-10）。当新块以某个 x 坐标置于轮廓顶端，其消耗分期 $O(1)$ 的时间确定它的 y 坐标。所有 B^* 树代表的打包必须是紧凑的，这样不会有单独块在不产生重叠的情况下下移。因此，B^* 树可能无法代表最小线网长度打包。如图 4-9 所示，如果 C 块与块 B 间有紧密连接，那么 a 中的打包比 b 中的打包的线网长度更短，但 a 不是紧凑打包所以它不能用 B^* 树代表。

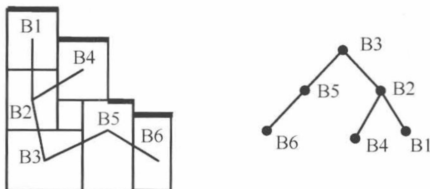


图 4-10　一个打包及其 B^* 树表示：该打包的轮廓用粗线表示

4.3.2　不同表示法的分析

在之前的章节中我们介绍了几种典型的 2D 布局规划表示方法。通过保存每芯片层的表示矩阵并允许不同层间的块交换，它们每个都可以扩展以解决含 2D 块的 3D 布局规划问题。但这些表示方法的解空间大相径庭。有些打包无法在某些表示法中表示。有些表示法实际上捕捉了与布局规划完全一致的集合。因此，我们从几

个角度分析表示法。

4.3.2.1　复杂度

基于表示法，我们需要扫描步骤来构建块的打包，我们称该步骤为布局规划构建。布局规划表示法一直用布局规划构建的算法复杂度和编码配置的总数衡量。本章参考文献［38，29］中讨论了多种布局规划表示法的算术特征并在这里简要总结。镶嵌布局规划的准确数目由 Baxter 数[38]给出，其可表示为

$$B(n) = \binom{n+1}{1}^{-1}\binom{n+1}{2}^{-1}\sum_{k=1}^{n}\binom{n+1}{k-1}\binom{n+1}{k}\binom{n+1}{k+1}$$

切片布局规划的准确数目在块数大于 1 时是超级卡塔兰数（Super Catalan Number）的两倍。超级卡塔兰数可表示为

$$A_0 = 1;\ A_1 = 1;$$
$$A_n = (2(2n-3)A_{n-1} - (n-3)A_{n-2})/n$$

图 4-11 示出了不同结构组合的准确数。我们可以看到 SP/TCG 的组合数随块的数目增加十分迅速；代表紧凑结构的 O 树 B* 树拥有最少的组合数。

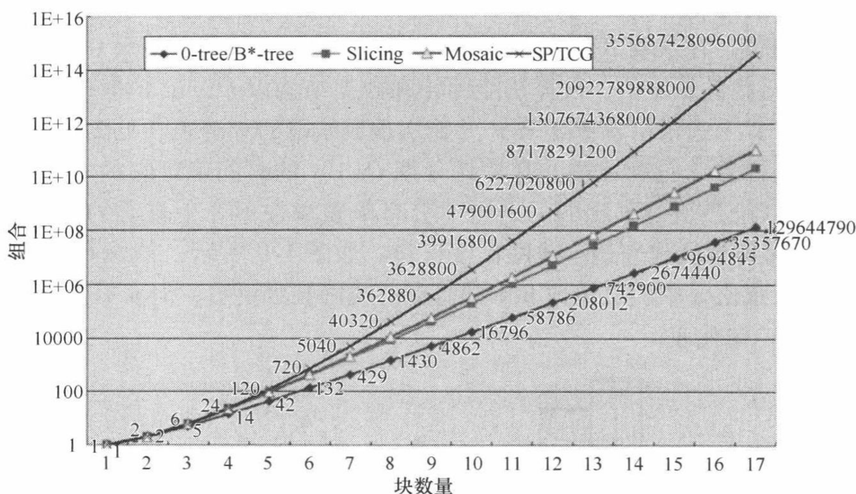


图 4-11　不同结构的准确组合数目（注意 Y 轴的组合数目采用的是指数坐标）

表 4-1 示出表示法之间在解空间、打包时间、打包类别上的比较。

有些表示法虽然不同，但在解空间上是等价的。解空间定义了表示法基本的表达范围，该范围可能直接影响解的质量。如果表示法共享相同的解空间，令移动集合相同，那么它们的差异就仅在于运行时间（某些移动会稍快或稍慢）。本章参考文献［14］表示因为共享相同的 $(n!)^2$ 解空间并捕捉相同的布局规划集合，序列对和 TCG 是等价的。每个序列对对应一个 TCG，反之亦然。尽管 TCG 和 SP 等价，它们的特征和相关操作差异很大。SP 和 TCG 都是很灵活的表示法并通过构建约束图衡量其打包代价。然而，像大多数现存的表示法，块间的几何关系对 SP 的操作

并不透明（即一个操作对模块关系的影响在打包前不清楚）；于是我们需要用每次移动后的草图构建约束图来衡量打包损失。该缺陷使 SP 更加难以覆盖所需的解空间和处理带约束的布局问题（例如边缘模块，预布局模块）。与 SP 相反，块间的几何关系对 TCG 及其操作是透明的，使之易于覆盖所需的解空间。此外，TCG 在操作中支持增量更新并同时保存边缘模块和表示法中模块的形状和相对位置的信息。

表 4-1　不同表示法对比

表 示 法	解 空 间	布局构建的复杂度	移　　动	打 包 种 类
NPE(SST)	$O(n!\ 2^{3n-3}/n^{1.5})$	$O(n)$	$O(1)$	片型
SP	$n!^2$	$O(n\log\log n)-O(n^2)$	$O(1)$	通用型
BSG	$n!\ C(n^2,\ n)$	$O(n^2)$	$O(1)$	通用型
O 树	$O(n!\ 2^{2n}/n^{1.5})$	$O(n)$	$O(1)$	紧密型
B* 树	$O(n!\ 2^{2n}/n^{1.5})$	$O(n)$	$O(1)$	紧密型
CBL	$O(n!\ 2^{3n-3}/n^{1.5})$	$O(n)$	$O(1)$	镶嵌型
TCG	$n!^2$	$O(n^2)$	$O(n)$	通用型

O 树和 B* 树使用单个树来表示水平紧凑打包，但与树的按位级实现不同。O 树使用具有任意顶点度数的根序树，而 B* 树使用二叉树。因此，它们共享大小为 $O(n!\ 2^{2n}/n^{1.5})$ 的相同解空间并捕捉相同的布局规划集合。与其他表示法相比，B* 树和 O 树的解空间相对较小。然而，它们仅代表部分拓扑信息而且需要所有块的尺寸以描述准确的布局规划。

4.3.2.2　冗余

冗余意味着代表某个布局规划的表示超过一个。表示的冗余会在多种找寻步骤中浪费步骤。实际上，如果我们考虑退化的情况（见图 4-12），大多数表示法对于该打包将有最少两种表示。以 NPE 为例，虽然有两种分块选择，但不论先采用哪种分块所得到的切片树仍是偏移的。而本章参考文献［7，38，40，39］中的大多数的工作将冗余情况视作特殊情况并假定交叉块被小段距离隔开使得块间拓扑关系能被固定。因此，我们不认为退化情况的多重表示是冗余表示。但即使不考虑退化情况，在某些表示中仍存在冗余——有些可以修正的，有些则不可避免。

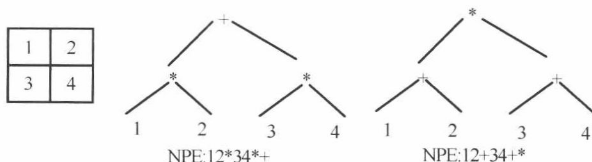


图 4-12　对应两个 NPE 表的退化情况

对于角块表，任意位表 T 及其他两表 S 和 L 能表示镶嵌打包。表 T 由长度不超过 $2n-3$ 的位表组成。T 的长度会随着打包结构动态变化；大多数的打包不需要表中 $2n-3$ 的全长。但如果我们在表示中为表 T 指定固定长度，一些 CBL 具有相同 S 和 L，但表 T 的尾部不同可表示相同的包。为弥补该不足，可以在打包过程中记录表 T 的有效长度，这样在优化过程中就能控制冗余移动的概率。如图 4-13 所示，如果将 T 的长度固定为 $2n-3=5$，那么图 4-13 中的两个表都代表相同的打包。原因在于 T 的有效表仅为 {0 0 0}，意味着块 2、3 和 4 仅覆盖 1 个块。因此，T 的有效长度为 3；然而，如果我们考虑到该信息，那两个表就是相同的。

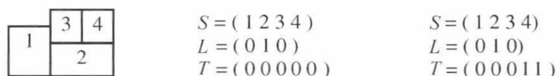


图 4-13　CBL 表示中的冗余：T 的合法长度是 3

在 SP 中，两个序列 Γ + 和 Γ -，分别从左上到右下和从左下到右上排列了所有的块。当两个块相对位置都是之上和之右，那么它们在 Γ + 中的相对位置有多种选择（见图 4-14 中的块 D 和 E）。同理，如果两个块相对位置都是之下和之左，那么它们在 Γ - 中的相对位置就有多种选择。该冗余表示导致了布局规划到表示的一对多映射。

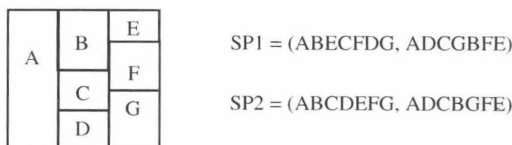


图 4-14　对于相同打包的冗余 SP 表示

考虑布局规划的表示，NPE[34] 是对于切片布局规划的非冗余表示法。TBS[39] 和 Q 序列[42] 是两种对于镶嵌布局规划的非冗余表示法。然而，没有对于总体布局规划的非冗余表示法。尽管所有总体布局规划可通过在 TBS 中插入空闲空间实现，但描述插入哪个空闲空间的信息是不一致的。因此，TBS 不容易扩展成为完整描述总体布局规划的简便表示法。

4.3.2.3　3D 设计的适应性

为了将 2D 布局规划表示法扩展以处理含 2D 块的 3D 布局规划，要构建 2D 表示矩阵（2D 矩阵），其中每个矩阵代表位于一个芯片层的用任意 2D 表示法表示的所有块。有两种实现层指定的方法：①在打包优化之前，考虑某些层间约束或目标的情况，把块指定到层，然后在打包优化过程中保持层指定关系不变；②初始化层信息，随后在打包优化过程中在层间交换块。第一种方法可能会限制解空间并失去最终结果的最优解，但能简化问题并使层间约束更容易满足。第二种方法更灵活并将有可能得到多个目标间更好的折中。因此，在本章中，我们使用第二种方法，即

使用层指定和每层布局规划同步确定的方法。

相比 2D 布局规划，含 2D 块的 3D 布局规划需要考虑更多的问题，如热分布、垂直相对位置约束以及热过孔插入。由于 B^* 树和 O 树仅代表紧凑打包，有可能捕捉不到最小线网长度解和最优温度。出于热分布的考虑，由于块间的白空间有助于隔离热块并可用于热过孔插入，紧凑打包不是必需的。同时为处理物理关系约束，如对齐约束，块间的几何关系非常有用。我们比较的典型的 2D 表示法来说明仅含 2D 块的 3D 布局规划的脉络。

SP 和 TCG 都可表示总体打包，复杂度为 $O(n^2)$。冗余典型地视为对序列对和 TCG 的限制。序列对表示法较简单而且衡量移动的时间更短，但是块间的几何关系不如 TCG 中的清晰，因而 TCG 易于扩展到处理物理约束。

如 CBL 基于空间的表示法也是一个不错的选择，因为其中块可以在空间中移动而不影响表示和拓扑关系，因此本地增量改善将更容易。CBL 表示法可在线性时间内衡量，而且有比 SP 和 TCG 更小的解空间，但它仅能表示镶嵌打包。

因此，通过不同复杂度和灵活性的折中，多种表示法可用于含 2D 块的 3D 布局规划。在 4.5.2 节中采用了 TCG 表示法，以一个桶形结构编码 z 轴相邻信息，这方法称为桶和 2D 矩阵的组合（CBA）[3]。

4.4　含三维块的三维布局规划表示法

与 2D 打包相似，3D 立方打包也分为两大类：切片和总体非切片。在总体 3D 封装中，也有一个子集称为 3D 镶嵌打包，其包含所有切片结构和部分非切片结构。因此，本节中介绍几种典型的表示法：3D 切片树[2]、3D CBL[16] 和三元序列、五元序列[35]。

4.4.1　三维切片树

我们可以用与 x、y 或 z 轴垂直的面递归切割 3D 块得到切片结构（假定 3D 块的表面与 x、y 或 z 轴垂直）。切片布局规划可用定向二叉树表示，称为切片树（见图 4-13）。树的每个内部节点标示为 X，Y 或 Z。X 坐标表示对应的平面模块是由垂直于 x 轴的平面切割得到的。同理，Y、Z 分别代表 y 和 z 轴项。每个叶子节点对应一个基本 3D 块并用其名称表示。与 2D 切片表示法类似，偏移的 3D 切片树也能用于避免冗余。在 3D 的偏移切片树中，没有节点与其右子节点有相同的表示（见图 4-15）。

4.4.2　三维 CBL

3D 打包的拓扑是在 3D 块对之间相对关系的系统。采用的方式是如果块“a”的任意点均在“b”的任意点左侧，那么称“a”在“b”之左。“之右”、“之上”、

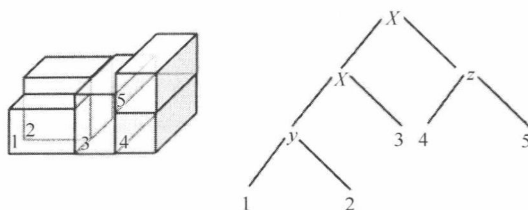


图 4-15 用偏移切片树表示的三维切片布局规划

“之下”、“之前”、“之后” 的关系近似地定义。与 2D 打包的镶嵌结构类似，3D 布局规划将总打包区域分割为有侧面的立方空间。每个立方空间含一个立方块。

因此，为了代表 3D 镶嵌布局规划的拓扑关系，每个立方块用一个立方空间表示并且空间在 x、y 或 z 轴方向互相覆盖。在从左下前角向右上后角的打包过程中，如果块 A 的空间覆盖块 B 的空间，那么块 A 的空间彻底地被块 B 的侧面及侧面的扩展面覆盖。如图 4-16 所示，每个空间的方向由其覆盖其他空间的方向定义。因此，如果一个新块 4 将插入图 4-16b 的打包，那么块 4 能在 x、y 或 z 方向上覆盖那些打包的块 {1，2，3}。新插入的块将定位在右前后角以使其可定义为角立方块。对于每个方向，因为其中的一些可能已经被之前打包的块覆盖，所以不是所有的打包块的空间可被覆盖。如图 4-16b 所示，块 1 已经被块 2 在 x 方向上覆盖；块 4 的新空间仅能在 x 轴方向上覆盖块 2、块 3 或同时覆盖两者的空间。因此，一个打包序列中未覆盖的块可定义为每个方向，其记录了当前可被覆盖的块。在图 4-16b 中，在块 4 插入前，z 轴方向上未覆盖的块表是 {1，2，3}，y 轴方向上未覆盖的块表是 {1，3}，x 轴方向上未覆盖的块表是 {2，3}。

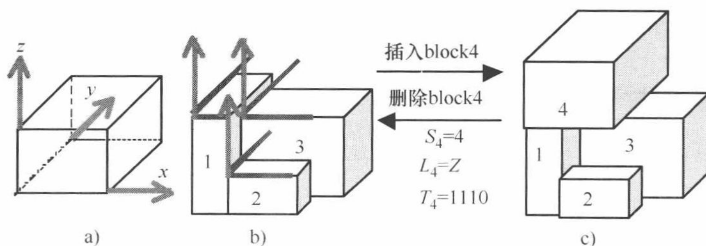


图 4-16 角立方块的过程：a）确立 x、y、z 方向；b）角立方块为 3 并解开的块表在 z 轴方向为 [1，2，3]；c）角立方块为 4，当 $T_4=1110$ 时其在 {1，2，3} 面上覆盖了 3 块；在 z 轴的未覆盖块表变为 {4} 并且对应 3D CBL：$S=\{1, 2, 3, 4\}$ $L=\{X, Y, Z\}$ $T=\{10\ 10\ 1110\}$

除了有覆盖方向和对应方向上的未覆盖块表，仍需要的信息是要覆盖哪个或哪些块以确定插入块的位置。假定某个打包序列中一个方向上的未覆盖块表为 $\{B_1, B_2, \cdots B_k\}$。如图 4-16c 所示，z 轴方向上的未覆盖块表是 {1，2，3}。如果块 4 的空间覆盖了块 1 的空间，块 4 的空间将同时覆盖块 2 和块 3 的空间。因此，为确

定插入块的位置，被该块空间覆盖的块数记录在未覆盖块表中。随着新块的插入，未覆盖块表应该动态更新。最后 m 块 $\{B_{k-m+1}, \cdots B_k\}$ 在此方向不再可被覆盖，于是，更新的未覆盖表在块 B 插入后应为 $\{B_1, \cdots B_{k-m}, B\}$。

因此，与插入的角块 B 的打包过程相关的信息应包括：块名称、覆盖方向和在未覆盖表中被 B 覆盖的块数。为优化过程利于新解的生成，二进制序列 T_i 用于记录在未覆盖表中覆盖的块数，其中“1”的数目对应覆盖的块数。每个“1”串以“0”结束来将其与下一个块的记录分隔开来。给一个 3D 打包，可以得到块名称的序列 S、方向的列表 L 和覆盖信息表 $\{T_2, T_3, \cdots, T_n\}$。三元变量 (S, L, T) 组成了 3D CBL（见图 4-16c）。图 4-17 按步骤示出了打包实例。

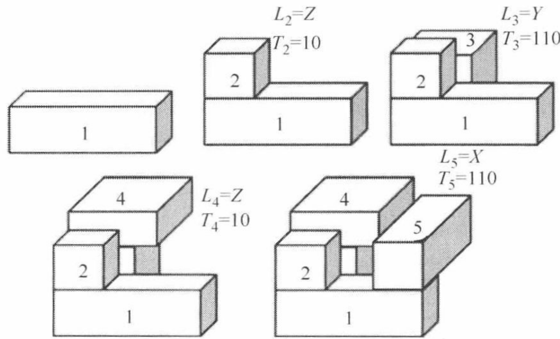


图 4-17　打包过程：$S=\{1\ 2\ 3\ 4\ 5\}$；$L=\{Z, Y, Z, X\}$；$T=\{10, 110, 10, 1110\}$

4.4.3　三元序列

三元序列（ST）[35] 是顺序三序列块标，其由 2D 打包使用的序列对扩展而来。三元序列（ST）表示为 ST（Γ_1，Γ_2，Γ_3）。与 SP 相似，ST 也有其解码规则以表示块间的拓扑关系。

(…a…b…，…a…b…，…a…b…)→b 在 a 之后
(…a…b…，…a…b…，…b…a…)→b 在 a 之左
(…a…b…，…b…a…，…a…b…)→b 在 a 之右
(…a…b…，…b…a…，…b…a…)→b 在 a 之下
(…b…a…，…b…a…，…b…a…)→b 在 a 之前
(…b…a…，…b…a…，…a…b…)→b 在 a 之右
(…b…a…，…a…b…，…b…a…)→b 在 a 之左
(…b…a…，…a…b…，…a…b…)→b 在 a 之上

给定 ST，3D 打包的实现如下：将表示法解码到 RL、FR 和 AB 拓扑的系统。随后，构建 3 个约束图 G_{RL}、G_{TR} 和 G_{AB}，都与 2D 打包类似。然后，到每个顶点的最长路径长度定位了对应的箱体，即左下前角的 (x, y, z) 坐标。

图 4-18 是 3 个块及其对应 ST 的打包实例。由于在 3 个块间有空洞，图 4-18 中的打包不是 3D 镶嵌打包，而且不能由 3D CBL 表示。

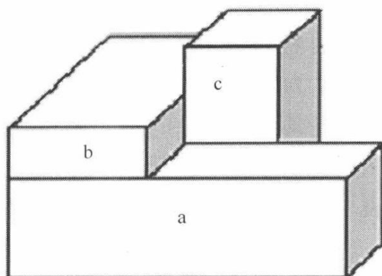


图 4-18 含空闲空间的打包可以用三元序列（bac，acb，abc）表示

与 3D CBL 相似，3D 块对之间的相对关系有之左、之右、之前、之后、之上和之下。3D CBL 和 ST 给每个箱体对准确的直接关系约束以使约束保持可转移度。换句话说，如果在对间存在间接关系约束，与直接关系约束并无不同。如图 4-19a 所示，a 在 b 之下。同理，b 必须被约束到 d 之下。尽管 a 不需要约束到直接位于 d 之下，a 也通过 c 间接的约束到位于 d 之下。同理，a 通过 c 间接地约束到 d 之前。于是，a 间接地约束到 d 之前之下，并且 a 和 d 的对有两个间接的相对关系约束。存在需要在一个块对上包含两个或三个间接关系约束的 3D 打包。这些 3D 打包称为“β 型”。众所周知，β 型的 3D 打包不能用 ST 或 3D CBL 表示[16]。

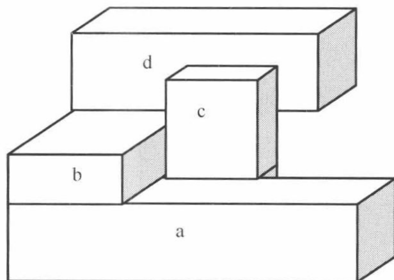


图 4-19 β 型 3D 打包，其中 a 块和 b 块具有两种非直接联系

因此，提出了表示所有 3D 打包的 5 个序列的系统，表示为 $Q=(\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5)$ 并称为五元序列（Squin）。从五元序列构建打包的算法如下：

- 第一步：构建左右约束图 GRL 来代表 Γ_1、Γ_2 的 RL 拓扑关系。遵循的原则与序列对相似，但不限于左右关系：

（Γ_1：<…，a，…，b，…>；Γ_2 <…，a，…，b，…>）→a 在 b 之前㊀

㊀ 疑原书误，应为之左。——译者注

由（Γ_3，Γ_4）构建出前后约束图 G_{FR}，遵循的规则如下：

（Γ_3：<…，a，…，b，…>；Γ_4 <…，a，…，b，…>）→a 在 b 之前

● 第二步：确定 G_{RL} 和 G_{FR} 中的最长路径以使每个块定位在其 $x-y$ 坐标。如果在投影到 $x-y$ 平面中重叠则称为两个块 $x-y$ 重叠。

● 第三步：构建上下约束图 G_{AB} 如下。对每个块对，加入从 a 到 b 的边并当且仅当 a 和 b 是 $x-y$ 重叠的并且 Γ_5 <…，a，…，b，…>。

● 第四步：通过 G_{AB} 中的最长路径确定 z 坐标。

五元序列可表示所有 3D 打包已被证明。从五元序列构建打包的算法的复杂度是 $O(n^2)$。

4.4.4　多种表示法的分析

3D 打包问题比 2D 打包复杂得多。我们从两个角度分析多个表示法：复杂度和含 3D 块的 3D 布局规划的灵活度。

1. 复杂度

表 4-2 给出了几个 3D 打包表示法的特性。切片树代表的 3D 打包比 3D CBL 少，3D CBL 代表的 3D 打包比 ST 少。ST 代表的 3D 打包比 Squin（五元序列）少（3D 切片树⊂3D CBL⊂ST⊂Squin）。Squin 可以代表任意 3D 打包。

表 4-2　一些 3D 打包表示法特性

表示法	布局构建的复杂度	移动的复杂度	打包种类	解空间
ST	$O(n^2)$	$O(1)$	部分通用型	$n!^3$
Squin	$O(n^2)$	$O(1)$	全片型	$n!^5$
3D 切片树	$O(n)$	$O(1)$		$O(n!\ 3^{n-1}2^{2n-2}/n^{1.5})$
3D-subTCG	$O(n^2)$	$O(n^2)$	部分通用型	$n!^3$
3D CBL	$O(n)$	$O(1)$	镶嵌型	$O(n!\ 3^{n-1}2^{4n-4})$

如果某些分块面在同一条线相遇，我们称此情况为退化拓扑。如果将退化拓扑视为一种特殊情况，那么[㊀]可以通过将一个面移开一小段距离分开它们以使块间的拓扑关系唯一。在此假设下，偏移 3D 切片树是非冗余表示。随着 T 表信息的动态更新，3D CBL 也可以非冗余地表示 3D 镶嵌打包。但 ST 和 Squin 因其转移特性而具有冗余。当两块的相对位置都为右上或都为右下等情况时，它们在表中的相对位置有多种选择。该冗余表示引发了布局规划到表示法的一对多映射。

2. 含 3D 块的 3D 布局规划的灵活度

含 3D 块的 3D 布局规划的问题需要为每个块在候选池里选择最优的配置。

㊀ than 应为 then，原书误。——译者注

同时由于一些设计约束，如 Z 高度约束，在优化过程中是强制性的。因此，如果含 3D 块的 3D 布局规划表示法可以灵活处理该要求的话，将会更好。基于实现，3D CBL 中从表到打包的转化可以在线性时间内逐次地从左下到右上进行。与基于图的表示法或序列家族相比，通过动态修复违反情况来处理约束会更容易。在以下的章节中，一种基于 CBL 表示法的试探法可在打包过程中用于修复违反 Z 高度的约束。该方法确保了最终结果的可行性并改善了收敛过程。基于 3D CBL 的立方布局规划构建是 $O(n)$ 次的，其中 n 是块数。但 3D CBL 主要的缺点是打包解空间大大小于 ST 和 Squin。因此，任意 3D 打包表示法均可用于 3D 布局规划。

4.5 优化技术

由于2D 和 3D 矩形打包问题是 NP 困难的，大多数布局规划算法基于模拟退火和遗传算法的随机组合优化技术。但最近有些研究关注定性方法从而使解析算法用于 3D 布局规划。

4.5.1 模拟退火

现今当穷举法代价太高时，随机优化方法用于大量的应用场合。3D 布局规划的目标是通过搜索由特定表示法表示的解空间，最小化给定的成本函数。通常，成本函数描述了芯片面积、线网长度、最大片上温度或其他因子的组合。

在本节中将介绍模拟退火方法在 3D 布局规划中的应用。一般来说，模拟退火法是由蒙特卡罗方法用于检查体系统状态和冻结状态等式的方法归纳得到的[20,8]。作为最常用的随机优化方法之一，模拟退火成功地应用于许多 VLSI 版图领域的问题。算法模拟了熔化点附近的退火温度，随后缓慢冷却以使其结晶为高度规则的状态。消耗在每个温度的时间要足够长以达到热平衡。图 4-20 示出了基于模拟退火方法的优化步骤。

4.5.2 基于 SA 的含二维块的三维布局规划

随着加入的 Z 方向，堆叠结构显著扩大了解空间。因此，一些基于 SA 的 3D 布局规划方法[10,36]提出了分级结构，其中层指定和布局规划连续实现。每个块的层号在模拟退火过程中固定。虽然这些方法减小了问题的复杂度，但它们也可能由于在优化过程中限制层指定而失去了优化性。这里介绍一种平面设计结构[3]，其中层指定和每层的布局规划同时确定。因此，块可在搜索过程中从一层移到另一层。结合之前章节介绍的表示法，可在含 2D 块的 3D 布局规划中应用 SA 优化方案。要设计一种高效的 SA 方案，下面几个问题很关键：

1）解的表示法：由于每层的打包可以用 2D 表示法表示，故多层打包可以用

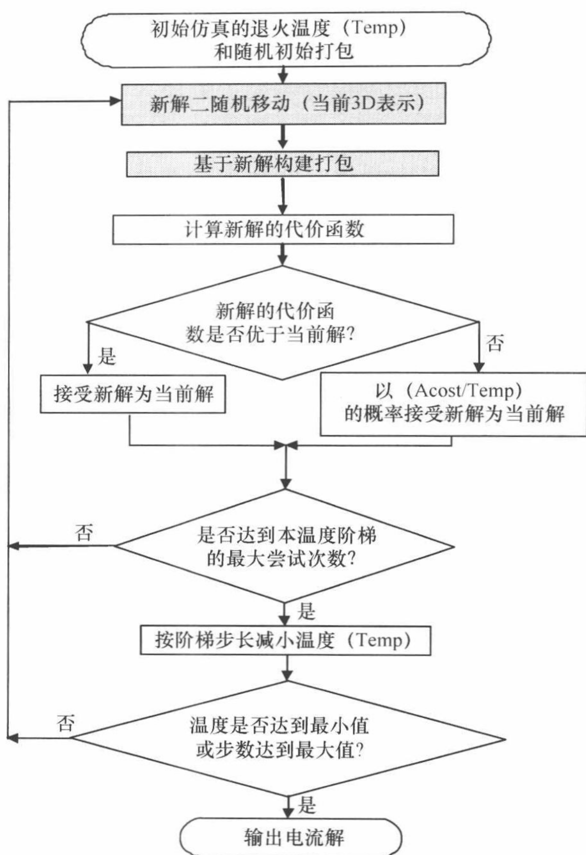


图 4-20　仿真的退火方法的流程图

2D 表示法矩阵表示并且通过将块移入每层或在层间交换实现解的微调。但为了克服缺少不同层间块的相对位置信息引入的限制，可将 z 方向相邻信息使用附加的桶结构编码。在每个桶 i 中，存储了与桶交叉的块的索引；不论块位于哪层之上，索引集均定为 IB(i)。同时，每个块 j 存储了对所有与块交叠的桶的索引；索引集定为 IBT(j)。因此，提出了桶和 2D 矩阵（CBA）的组合；由两部分组成——一种用于表示每层的 2D 布局规划表示法，一种存储块间垂直关系的桶结构。在本章中选择 TCG 来表示每层上的 2D 打包。

2）冷却步骤：整个冷却步骤包括初始温度的设定、冷却函数、终止温度。该过程依赖问题的大小和问题的特性。

3）解的微调：以 CBA 表示法为例。共有 7 种 CBA 上的操作，如下所示：

① 旋转，旋转块；

② 交换，同层间交换块；

③ 反转，交换同层中两块的相对位置；

④ 移动，将块从某块的一侧（如上方）移动到其另一侧（如左方）；

⑤ 层间交换，在不同层间交换块；

⑥ z 相邻交换，交换在不同层间但互相邻近的块；

⑦ z 相邻移动，将块移动到另一层与当前位置邻近的位置。

4）成本函数：每生成一次块配置，就会求一次优化目标和约束的加权成本。成本函数可表示为

$$\text{Cost} = \alpha W_L + \beta A_{rea} + \gamma N_{via} + \theta T$$

式中，W_L 是用半周长模型估算的线网长度；A_{rea}是所有层最大高度和宽度的乘积；N_{via}是层间过孔的数目；T 是最大温度。在 3D 设计中，片上温度极高，导致必须计算封闭的温度/漏电功耗反馈环才能准确估计或优化这两者。

4.5.3 基于 SA 的含三维块的三维布局规划

基于模拟退火的含 3D 块的立方打包过程与含 2D 块的 3D 布局规划相似。但我们这里研究的含 3D 块的 3D 布局规划问题不止关乎块的位置，还包括块的配置。与之前基于模拟退火的布局规划方法不同，在打包过程中，“块”配置的选择被动态的集成。

因为表示法是模拟退火的关键问题，我们以 3D CBL 为例介绍基于 SA 方法的过程。由于根据不同的分块方法使候选块在尺寸、延迟、功耗和层数上的变化，块的配置信息可在优化过程中选择。因此，为了给块选择最可行的配置，定义了新操作“可替换选择”来创建新解。

可替换选择如下：

1）在多个候选块中随机选择块 i；

2）在候选表中随机选择一个可行的候选；

3）根据候选的尺寸更新块 i。

生成邻接解所用的移动基于任意一种以下操作：

1）随机交换 S 中块的顺序；

2）随机在 L 中选择位置并改变方向；

3）随机在 T 中选择位置，使“1”变为“0”或使“0”变为“1”；

4）可替换选择。

多种组分候选极大地扩展了解空间。特别对于一些层数约束，部分解是不可行的。因此，建议使用试探的方法来加速搜索进程。

成本函数也使用加权的面积、温度和线网长度组合，其可表示为

$$\text{Cost} = w_1 \times A_{rea} + w_2 \times T_{emp} + w_3 \times W_{ire}$$

在块的当前布局规划下，A_{rea}（面积）是布局规划的总面积。T_{emp}（温度）对应基于温度仿真器的最大片上温度。系数 w_1，w_2 和 w_3 用来控制每个组分的不同加权。

在3D微结构设计中，芯片层的数目经常以约束的形式给出。为了处理层数约束，传统方法是在成本函数中处罚违反情况。然而，该方法不能保证最终结构的可行性，并将减慢优化的收敛速度。在3D CBL表示法中，块打包在序列中。因此，块或CBL表可在打包过程中动态的改变。如果某些块超出了层数约束，那么这样的违反可通过降低块或改变块的方向修复。我们采用以下的步骤修复违反情况：

1）尽可能多地保留拓扑结构，首先通过选择一个较低的候选方案努力改变块实现。

2）如果通过改变候选方案不能修复违反，就努力修改3D CBL表来实现一种可行的打包。如果块B覆盖了z轴方向上之前的块，意味着块B将置于打包块的顶端，同时如果块B超过了层数约束，我们可以将覆盖方向改变到x或y方向以使块B置于之前块右侧或后侧。但如果块B的z位置仍然过高，我们可以通过增加T_B中“1”的数目将块B动态地移动到更低的位置。因为T_B表示在L_B方向上被B覆盖的块的数目，所以当我们增加T_B中“1”的数目时块B将被移动到更低的块。该过程将持续到块B满足层数约束为止。给定设计Z_{con}中的层数，就会将CBL表扫描以从左下前角向右上后角打包块。打包的块B的左下前角坐标为（x_B，y_B，z_B），对应的实现为C_j^B。于是过程可描述为

修复违反算法

输入：块B超出层数限制：$z_B + z_j^B > Z_{con}$；3D_CBL和块B的候选表。

输出：采用B的新候选选择c^B的新3D_CBL；

```
If z_B < Z_con
    For 候选表B中的候选 c_j^B
        If z_B + z_j^B ≤ Z_con
            选择此候选 c^B = c_j^B 并更新B位置；
            返回；//违反通过改变候选修复
        EndIf;
    EndFor;
    选择具有最低z高度的候选并更新B的信息；
    If L_B = Z;//在z方向覆盖之前块
        改变L_B到x或y方向并更新B位置；
    EndIf
    While(z_B + z_j^B > Z_con)
        在T_B^LB中增加“1”的数目,其意味着在L_B方向上被B覆盖的块增加。
        更新B的位置；
    EndWhile
EndIf
```

极端的情况是块 B 移动到底部（$z_B=0$）。候选表应在每个块的 z 高度小于 Z_{con} 的约束下构建。如果 $z_B=0$，块 B 将不会超过层数约束。因此，我们的算法保证了结果的可行性。

4.5.4 解析方法

大多数布局规划算法都基于模拟退火技术。但随机优化方法基本上运行时间较长，无法有效减小问题的大小。这里，解析的方法提供相对稳定和可按比例缩小的 3D 布局规划优化技术。在标准单元的布局算法中解析方法得到广泛的应用[4,5,21]（将在第 5 章中详细介绍）。但在宏单元的布局规划上，块尺寸和块形状的异质性使问题复杂化。在优化过程中的一个小变化能引起最终合法打包的巨大变化。在随机优化方法中，块的拓扑关系由表示法描述以保证块互相不重叠。而用数学计算来以线性的方法规定块间的不重叠约束是困难的；因此，移除块间重叠情况的合法化在大多数解释方法中是必需的。在本节中将简要介绍处理温度敏感的含 2D 块的 3D 布局规划的力导向方法，此方法在本章文献［41］提出。含 3D 块的布局规划与此类似，并且以下介绍的方法可以容易地扩展到处理 3D 块。

从布局的解开始，需要从连续空间翻译到分开的、层指定并合法化的空间的解。因此解析方法有 3 个过程：全局布局、层指定和合法化，如图 4-21 所示。

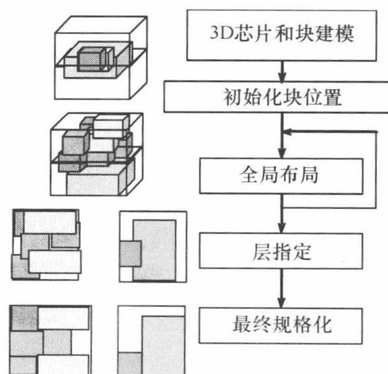


图 4-21 三维力导向布局规划流程

1. 全局布局

有许多在连续区域优化单元位置的数学方法（将在第 5 章详细介绍），这里，我们以基础的力导向方法为例进行介绍。力导向算法模拟了质点附在弹簧上并且它们的移动遵循胡克定律的机械问题。同质立方箱体结构引入两种力，3D 空间中的填充力和热力来消除重叠并减小布局峰值温度。

• 填充力：填充力用于消除块间重叠并将它们平均分布在 3D 布局空间区域中。其驱动布局通过在 3D 空间中将块从高密度区域推开并推向低密度区域来去除

重叠。箱体密度定义为覆盖该箱体的块数之和。每个箱体的填充力等于其箱体密度。块收到的填充力等于所述单元覆盖的按比例分配的箱体的填充力。

- 热力：（第 3 章中介绍的）热模型得到了布局的热梯度。我们将乐于把块（产生热的）移动到高温的区域之外。该目标可通过使用热梯度以确定块上热力的方向和大小来实现。给定块的填充力和热力的计算是对箱体在每个树的层次上获得的独立力求和。要考虑从一个箱体及其最近邻居发出的力。大块占据多个箱体，结果是其受到比小块更大的力。

2. 层指定

在优化连续 3D 空间的布局之后，块必须指定到分开的 IC 层上。在以上方法中，每个块建模为 3D 矩形并可在连续 3D 空间中自由移动。层指定将块连续空间移动到分离空间，强制每个块占据仅一个 IC 层。力导向的方法努力逐步将块平均分布在空间中。层指定基于块在 z 轴的位置，其通过力导向方法得到的当前布局获得。图 4-22 示出了 3 个块的层指定过程。

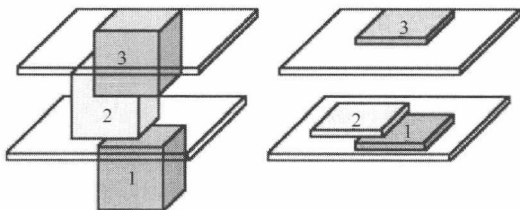


图 4-22　层指定

3. 最终合法化

在之前章节介绍的全局布局之后，我们得到了带有少量重叠残留的多层打包解。为得到可行的布局，合法化策略微调了解以产生无重叠的打包，并同时尝试保持块间初始的拓扑关系。合法化问题的定义可描述如下：构建重叠块间的拓扑关系使块错位最小化。根据其位置将块按从芯片左下角到右上角排序，可得到粗略的拓扑序列。如图 4-23 所示，序列中块 a 位于块 b 之前，它们相互重叠。我们必须确定块 b 是否在块 a 的右侧或上方并选择最优的方向。在之前章节引入的 2D 表示法可用于代表块间的拓扑关系。此外，块可在合法化过程中旋转，这同时有助于控制重叠移除引发的错位。由于块间的拓扑关系通过一些试探性规则确定，直接的合法化会产生相对于初始布局的巨大错位。那么可以设计一种后过程以进一步改善合法化的结果。随机方法可用于本地调整打包以使打包过程得到进一步的优化。

该力定向解析方法在线网长度优化方面比模拟退火更有效，在下一节中将说明。然而，这样有两个问题：①在层指定方面，满足箱体的密度约束不一定能得到合法的 3D 布局解。我们可考虑使用最近提出的力定向 3D 布局（在 5.4 节中说明并在参考文献［4］中详细讲解），其介绍了在虚拟器件层上为了保证分立 3D 层上

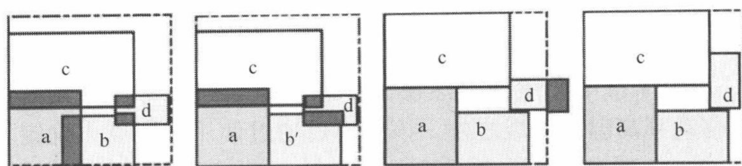


图 4-23 规格化过程

布局的合法性的密度约束。②此处给出的力定向解析方法仅应用于 2D 块，扩展到能处理 3D 块需要更多的研究。

4.6 多种三维布局规划技术的影响

在本节中我们总结了 2D 块和 3D 块构成的多种 3D 布局规划工具报道的实验结果。

4.6.1 含二维块的三维布局规划影响

尽管含 2D 块的 3D 布局规划已有大量的研究工作，这里我们仅总结两种代表性的使用通用示例集算法的结果：使用基于 CBA[3] 的模拟退火的 3D 布局规划器和力定向 3D 布局规划器[41]。所有的算法均在 MCNC 基准电路和 GSRC 测试通过。所有的电路均使用 4 个器件层。我们首先比较使用非热敏感的多种表示法的含 2D 块的 3D 布局规划算法的结果。如表 4-3 所示，线网长度使用半周长线网估计模型估算（HPWL）。与使用基于 CBA[3] 的模拟退火的 3D 布局规划器相比，在取所有基准电路平均值时，力定向方法减小了 4% 的面积，改善了 12% 的线网长度，并且在 CBA 所需运行时间的 69% 以内完成了程序的执行。

表 4-3 用 2D 块布局成的两个 3D 的面积和线长优化

电路	CBA[3]			力导向[41]		
	面积/mm^2	HPWL/mm	时间/s	面积/mm^2	HPWL/mm	时间/s
Ami33	35.3	22.5	23	37.9	22	52
Ami49	1490	446.8	86	1349.1	437.5	57
N100	5.29	100.5	313	5.9	91.3	68
N200	5.77	210.3	1994	5.9	168.6	397
N300	8.90	315.0	3480	9.7	237.9	392
	1	1	1	+4%	-12%	-31%

表 4-4 示出了 CBA 和力定向方法在优化面积、线网长度和温度上的比较。这里，每个块的功率密度定在 10^5 ~ $10^7 W/m^2$[3] 之间。一种空间可适应 3D 多层芯片封装热分析的软件包的扩展版本[37] 用作热分析模型，以用来衡量热分布情况。这里假定漏电流功耗是固定的。但依赖温度漏电流功耗模型可用于描述漏电流——温度反馈。读者们可参考本章参考文献［41］来获得详细信息。

表 4-4　优化区域、线长和温度时，CBA 和力定向方法比较

电路	CBA[3]				力导向[41]			
	面积/mm^2	HPWL/mm	温度/℃	时间/s	面积/mm^2	HPWL/mm	温度/℃	时间/s
Ami33	43.2	23.9	212.4	486	41.5	24.2	201.3	227
Ami49	1672.6	516.4	225.1	620	1539.4	457.3	230.2	336
N100	6.6	122.9	172.7	4535	6.6	91.5	156.8	341
N200	6.6	203.7	174.7	6724	6.2	167.8	164.6	643
N300	10.4	324.9	190.8	18475	9.3	236.7	168.2	1394
	1	1		1	-16%	-12%		-75%

图 4-24 示出了通过力定向方法得到的 4 层打包及其对应的功耗分布和热分布。高功率密度的块指定到底层以减小峰值温度。与基于 SA 的方法相比，解析方法更加稳定并能在更短时间内获得更好的结果。但基于 SA 的方法在处理附加的目标和约束时更加灵活。

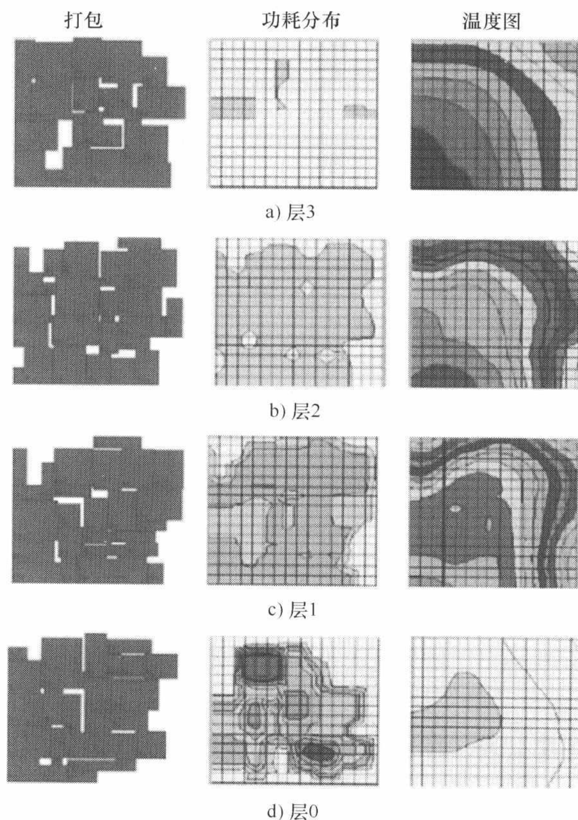


图 4-24　通过力导向方式得到的具有功耗分布和热剖面的 4 层打包

4.6.2　含三维块的三维布局规划的影响

大多数发表的含 3D 块的 3D 布局规划算法在用于本章参考文献［17］中提到的旅行者问题的基准电路上测试它们的算法。权重系数视为第三维度，并因此变为三维矩形打包问题。表 4-5 示出了 3 种算法的比较结果：3D CBL、ST 和 3D-TCG。对于这些结果，我们可以看到由表构建布局规划时 3D CBL 因其线性时间复杂度比其他两种算法运行更迅速。但受解空间的限制，3D CBL 的打包结果不如 3D-sub TCG 好，特别是对于较大电路的情况。

表 4-5　3 种算法结果比较：3D CBL、ST 和 3D-TCG

测　试	块编号	体积总量	ST		3D-sub TCG		3D CBL	
			死空间/%	运行时间/s	死空间/%	运行时间/s	死空间/%	运行时间/s
beasley1	10	6218	28.6	7.7	17.1	8.5	23.5	6
beasley2	17	11497	21.5	45.2	7.2	28.5	17.0	7
beasley3	21	10362	35.3	44.1	18.0	18.0	17.0	12
beasley5	14	16734	26.4	18.2	11.5	16.0	13.5	12
beasley6	15	11040	26.3	27.9	16.3	24.8	15.4	20
beasley7	8	17168	30.1	3.8	16.5	2.3	24.6	4
beasley10	13	493746	25.2	13.0	14.2	10.8	15.2	10
beasley11	15	383391	24.8	17.5	12.6	9.8	13.2	10
beasley12	22	646158	29.9	100.0	21.5	58.5	21.2	40
okp1	50	$1.24*10^8$	42.6	1607.2	28.4	387.3	29.1	202
okp2	30	$8.54*10^7$	33.2	285.3	22.3	73.8	27.0	57
okp3	30	$1.23*10^8$	33.1	280.7	23.0	70.6	26.3	56
okp4	61	$2.38*10^8$	42.8	791.3	27.3	501.9	28.6	320
okp5	97	$1.89*10^8$	57.7	607.8	35.8	565.9	36.2	340

为表示出热敏感的含 3D 块的 3D 布局规划的影响，衡量结果用于高性能超标量处理器[12,24]。表 4-6 示出了处理器使用的基本参数。由于每个关键元件有可表示为 3D 块的不同实现方式，打包引擎可成功打包块并在层数约束下为每个块选择最优实现。图 4-25a 示出了含 3D 块的两层打包布局规划的 3D 视图。面积为 $3.6\times3.6\text{mm}^2$。打包引擎在单层和两层块结构间选择。对于如 ALU、MUL 和二级缓存单元的块，选择使用单层实现。其余的块在两层上实现（我们使用立方块代表多层块）。

表 4-6　设计驱动程序的架构参数

处理器宽度	6 路乱序超标量体系结构，二整数簇
寄存器文件	128 项整数（两个已复制文件），128 项浮点处理器
数据缓存	8 KB 的 4 路集联，64B 块的大小
指令缓存	8 KB 的 2 路集联，32B 块的大小
二级缓存	4 组，每组 128 KB 的 8 路集联，128B 块大小
分支预测	8K 项 gshare 和 1K 项，4 路 BTB
功能单元	每两簇有两个 ALU 中断 + 一个乘法/除法中断；1 个 ALU 浮点运算和 1 个乘法/除法运算

图 4-25 还示出了两层设计中层的温度分布，其中顶层明显比底层热并有一个 90℃ 的热点。与散热体和热沉相接的底层比顶层温度低。尽管底层的功率密度比顶层高，但顶层到热沉的热阻更高。尽管硅是相对较好的热导体，但垂直方向的热传导受到了金属层、绑定金属和增加了的距离组合的负面影响。能改善从顶层到热沉的垂直热传导的热通孔可用来将热点的温度保持在给定的热阈值之下。

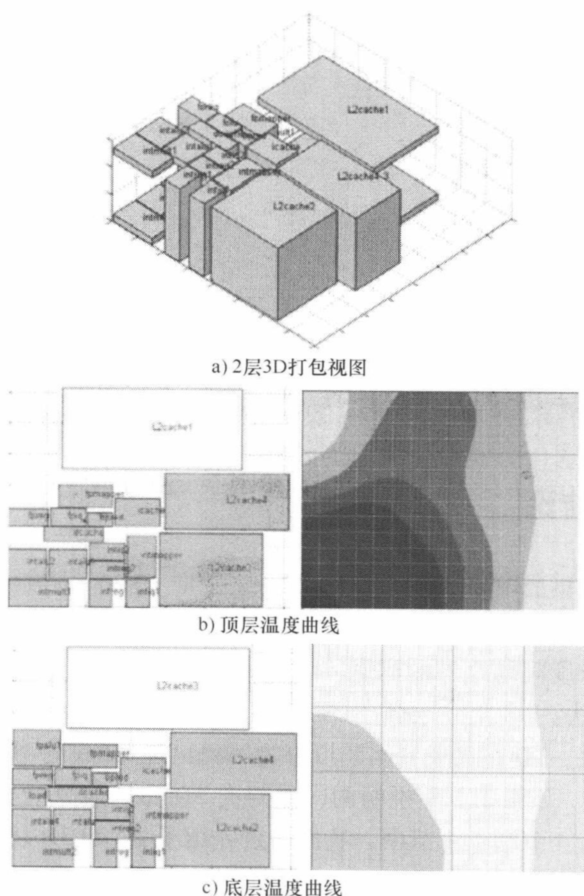


图 4-25　通过 3D 布局规划器得到的具有基于 3D CBL 表示的 3D 块的两层打包

图 4-26 示出了 2D 和 3D 结构的块工艺的温度比较。x 轴表示在 3 ~ 6GHz 频率范围内不同硅层数的不同配置。y 轴以℃的温度表示 3D 和 2D 块工艺及热通孔插入的结果。环境温度设为 27℃。如在本章参考文献［12］中示出的，多层 3D 块可比单层块节省 10% ~ 30% 的功耗。但温度严重依赖版图布局。为减轻热点，有必要经常性地让潜在的热点与其他潜在热点远离。即使单层块在这方面看起来比多层块更有优势，但 3D 打包引擎通过其根据块的热分布对块做的智能层选择克服了该问题。因此，我们可以看到对于两层或三层的设计，温度可以因多层块功耗的减小和可替换选择而降低。

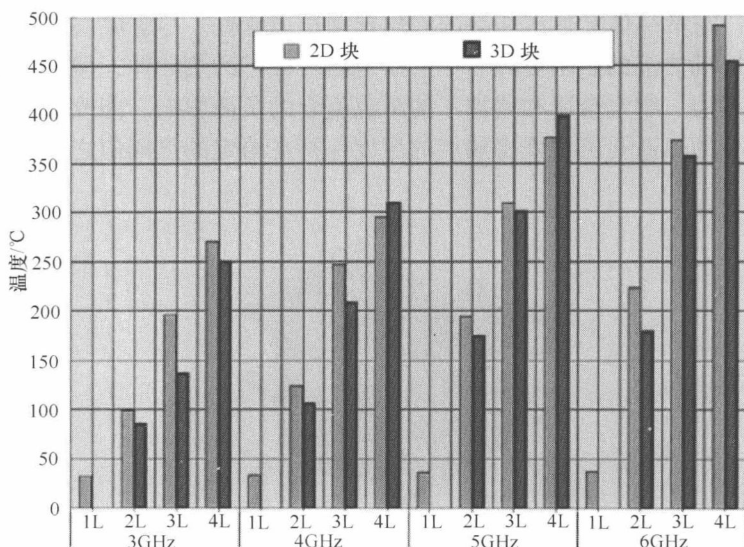


图 4-26　3 ~ 6GHz 和 1 ~ 4 层情况下 2D 和 3D 的温度比较

4.7　总结和结论

作为一种新的集成电路（IC）设计技术，三维（3D）集成的物理设计受到了在多层器件层结构上出现的设计方法和优化方向的挑战。这是在那些深亚微米工艺中出现的设计复杂度问题之外的问题。在本章中，我们介绍了含 2D 块和 3D 块的 3D 布局规划的算法。

根据块表示法，3D 布局规划问题可分为两类：含 2D 块的 3D 布局规划和含 3D 块的 3D 布局规划。如 4.2 节中说明的，这两类 3D 布局规划需要不同的表示法和优化技术。因此 4.3 节和 4.4 节中，我们分别介绍了 2D 和 3D 块的表示法。

由于含 2D 块的 3D 布局规划可表示为 2D 表示法矩阵，2D 布局规划算法通过在优化技术中引入新操作扩展到多层设计处理。在 4.3 节中，简要介绍了几种基本

的 2D 表示法，这些是 3D 布局规划优化的基本技术。不同表示法的分析展示了这些表示法的来龙去脉。在 4.4 节中，介绍了用来表示含 3D 块的 3D 打包的几种典型的表示法：3D 切片树、3D CBL、三元序列和五元序列。在 4.5 节中，在基于表示法的多种随机优化的简要介绍之外，还介绍了解析的方法。我们介绍了含2D/3D块的 3D 布局规划的经典优化方法——模拟退火法。通常用于标准单元布局的力定向方法的热敏感解析方法也在本节中予以介绍。

附录　折叠 3D 元件设计

最近的研究为包括 3D 缓存器[30,9,28]、3D 寄存器堆[31]、3D 算术单元[25]和 3D 指令调度器[26]在内的多种架构提供了块模型。为构建多层块以在机构设计中减小块内互连延迟和功耗，有两种主要的在多硅层上设计块的策略：块折叠（BF）和端口分块（PP）。块折叠意味着在 X 或 Y 方向上的块折叠可以潜在地减少该方向的线网长度。端口分块将结构的可访问端口置于不同的层。直觉上用于到单个块（如多端口缓存）的重复访问所需要附加的硬件可以分布在不同层，这样可以极大地减小在每层中的互连长度。例如，将这些策略用于类似缓存的块已经简要介绍了。对于所有其他的元件，如问题队列、寄存器堆，也可以进行类似的分析。

缓存通常在含固定结构的架构中出现。它们由大量的标签和数据矩阵组成。图 4-27示出了一个三端结构的单元。每个端口含位、位线、字线和每位 2 个晶体管。组成存储单元的 4 个晶体管比分配给端口的空间小得多。线间距一般是特征尺寸的 5 倍。对于每个额外的端口，在 X 和 Y 方向上的线网长度的增长速度是线间距增长速度的两倍。另一方面，由 4 个晶体管组成的存储部分的高度是线间距的 2 倍，宽度与线间距相等。因此，一个元件的端口越多，就有越多的硅芯片面积被端口占据。一个三端口结构的端口面积和单元面积的比率约为 18∶1。

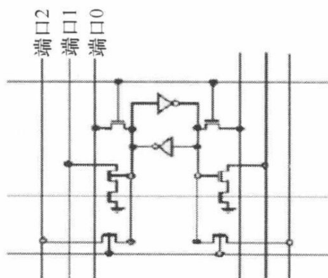


图 4-27　三端口 SRAM 单元

图 4-28a 示出了多个通过地址和数据总线相连的多个缓存标签和数据阵列的高层次视图。每个垂直和水平的线代表 32 位总线。假定该缓存上有两个端口，同时线路成对。缓存的元件可以简单地打开为子矩阵。CACTI[27,30]可用于探索不同子部分的设计空间并找到性能、功耗和面积的最优点。

1. 块折叠（BF）

块折叠有两种选择：字线折叠和位线折叠。在前者中，缓存子矩阵中的字线分开并置于不同芯片层上。字线驱动器也复制了。字线折叠的收获来自缩短了的从预

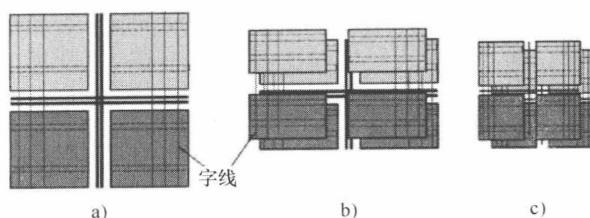


图 4-28 可替代缓存的三维块：a）2D 两端口缓存：图中线表示两个端口的输入/输出线网；b）字线折叠：仅 y 方向减小了。输入/输出端口翻倍；c）端口分块：端口置于两层。x 和 y 方向长度均减小了

解码器到解码器以及从输出驱动器到缓存边缘的布线距离。同理，位线折叠将位线置于不同层但需要复制传输门。调查显示，字线折叠在大多数情况下比理想位线折叠可实现更好的访问时间和功耗。这里，图 4-29 示出了使用字线折叠的结果。

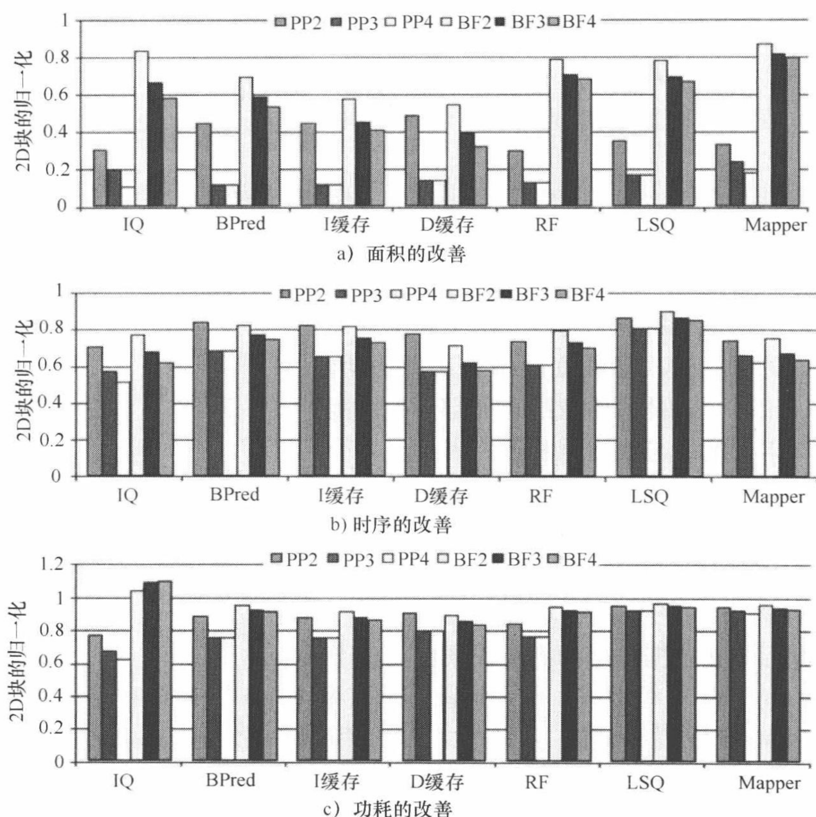


图 4-29 多层 F2B 设计的改善（PP2 指两层设计的端口分块，BF2 指两层设计的块折叠）

2. 端口分块

将端口分块并置于不同芯片有一个显著的优点，如图 4-28c 所示。在两层设计中，可将两个端口置于一层，一个端口和 SRAM 单元置于另一层。宽度和高度都减小了约一半，面积减小为 1/4。端口分块同时实现了垂直方向和水平方向线网长度缩短。这减小了总线网长度和总电容，并转换为访问时间的缩短和功耗的降低。端口分块需要通孔将存储器单元连接到在其他层上的端口。依靠工艺，通孔的直径也会影响尺寸。在我们的设计中，需要为每个所需通孔设置 $0.7\mu m \times 0.7\mu m$ 的空间。这与本章参考文献［30］中用来获取通孔电容和电阻的模型相同。

图 4-29 示出了不同分块策略对不同元件的影响。总结这些影响，我们可以得到：

1）端口分块对持续缩小面积比所有结构都更有效。原因是端口分块同时缩小了 x 和 y 方向的长度。

2）对于缓存，端口分块在层数增加时不提供功耗和时序的额外改善。原因是这些缓存没有足够多的端口。同时，晶体管层必须给通孔提供空间。另外，通过字线折叠，该趋势继续并随层数的增加持续改善。

3）平均来看，端口折叠比块折叠可节省更多面积。块折叠在缩减块延迟，特别是端口较少的块的延迟上更有效。端口分块在减小功耗上的性能更佳。

4）尽管多层块延迟和功耗相对于单层有所降低，但最差功耗密度由于层的堆叠将显著增加。因此，单个块的功耗减小不能保证热点的消除。热效应不仅依赖每个块的配置，还依赖布局的物理信息。

这两种方式的多种优势显示了对可根据独立布局规划约束灵活选择合适的实现工具的需要。选择合适的实现过程需要考虑线网流水线在内的物理信息。每个元件的最优的 3D 配置不一定会导向对整个系统最优的 3D 实现。在一些情况下，如 4 层芯片，如果元件选为 4 层块，其他块将不能置于其上方及相邻位置。此外，对所有其他高度互连的块来说，该块可能不够。因此，块间线网迟滞有可能增加并生成一些附加周期。另一方面，如果为该元件选择一个两层实现，尽管块内延迟不是最优，但由于其他块可以置于该块之上而使该块与其他块因垂直互连更短而紧密连接，最终使块间延迟受益。因此该元件 2 层实现的打包可能比其 4 层打包表现更优。此外，为有助于热效应，3D 块延时的缩小可提供延时迟滞以允许时序和功耗的折中。但该优化也将依赖物理打包结果提供的时序信息。

因此，为了应用 3D 块，决策不能简单地仅根据结构方面或物理设计方面做出。为了在 3D 微结构和物理设计之间实现协同优化，我们需要一个可以在打包优化过程中选择实现方式的真实的 3D 打包引擎。

致　谢

作者感谢以下支持：Gigascale Silicon Research Center，IBM DARPA 子协议，国家自然科学基金（美国）CCF-0430077、CCF-0528583，中国国家自然科学基金60606007、60720106003、60728205，清华基础研究基金 JC20070021，清华信息科学核技术国家重点实验室（TNList）交叉学科基金 042003011，是这些支持实现了本章中的大量成果。

参考文献

1. Y. C. Chang, Y. W. Chang, G. M. Wu, and S. W. Wu, B*-trees: An new representation for nonslicing floorplans, *Proceedings of ACM/IEEE DAC 2000*, pp. 458–463, 2000.
2. L. Cheng, L. Deng, and M. D. Wong, Floorplanning for 3D VLSI design, *Proceedings of IEEE/ACM ASP-DAC 2005*, pp. 405–411, 2005.
3. J. Cong, J. Wei, and Y. Zhang, A thermal-driven floorplanning algorithm for 3D ICs, *Proceedings of ICCAD 2004*, pp. 306–313, 2004.
4. J. Cong and G. Luo, A multilevel analytical placement for 3D ICs, *Proceedings of the 14th ASP-DAC*, Yokohama, Japan, pp. 361–366, January 2009.
5. B. Goplen and S. Sapatnekar, Efficient thermal placement of standard cells in 3D ICs using a force directed approach, *Proceedings of ICCAD 2003*, pp. 86–89, Nov. 2003.
6. P.-N. Guo, C.-K. Cheng, and T. Yoshimura, An O-tree representation of nonslicing floorplan and its application, *Proceedings of ACM/IEEE DAC 1999*, pp. 268–273, 1999.
7. X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, C. K. Cheng, and J. Gu, Corner block list: An effective and efficient topological representation of nonslicing floorplan, *Proceedings of IEEE/ACM ICCAD 2000*, pp. 8–12, 2000.
8. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi Jr, Optimization by simulated annealing, Science, pp. 671–680, May 1983.
9. M. B. Kleiner, S. A. Kuhn, P. Ramm, and W. Weber, Performance and improvement of the memory hierarchy of risc-systems by application of 3-D technology, *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, 19(4): 709–718, 1996
10. Z. Li, X. Hong, Q. Zhou, Y. Cai, J. Bian, H. Yang, P. Saxena, and V. Pitchumani, A divide-and-conquer 2.5-D floorplanning algorithm based on statistical wirelength estimation, *Proceedings of ISCAS 2005*, pp. 6230–6233, 2005.
11. Z. Li, X. Hong, Q. Zhou, Y. Cai, J. Bian, H. H. Yang, V. Pitchumani, and C.-K. Cheng, Hierarchical 3-D floorplanning algorithm for wirelength optimization, *IEEE Transaction on Circuits and Systems I*, 53(12): 2637–2646, 2007.
12. Y. Liu, Y. Ma, E. Kursun, J. Cong, and G. Reinman, Fine grain 3D integration for microarchitecture design through cube packing exploration, *Proceedings of IEEE ICCD 2007*, pp. 259–266, Oct 2007.
13. J. M. Lin and Y. W. Chang, TCG: A transitive closure graph-based representation for non-slicing floorplans, *Proceedings of ACM/IEEE DAC 2001*, pp. 764–769, 2001.
14. J. M. Lin and Y. W. Chang, TCG-S: Orthogonal coupling of P*-admissible representations for general floorplans, *Proceedings of ACM/IEEE DAC 2002*, pp. 842–847, 2002.
15. Y. Ma, X. Hong, S. Dong, Y. Cai, C. K. Cheng, and J. Gu, Floorplanning with abutment constraints and L-shaped/T-shaped blocks based on corner block list, *Proceedings of DAC 2001*, pp. 770–775, 2001.

16. Y. Ma, X. Hong, S. Dong and C. K.Cheng, 3D CBL: an efficient algorithm for general 3-dimensional packing problems, *Proceedings of the 48th MWS-CAS 2005*, 2, pp. 1079–1082, 2005.
17. F. K. Miyazawa and Y. Wakabayashi, An algorithm for the three-dimensional packing problem with asymptotic performance analysis, *Algorithmica*, 18(1): 122–144, May 1997.
18. H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, Rectangle packing based module placement, *Proceedings of IEEE ICCAD 1995*, pp. 472–479, 1995.
19. S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, Module placement on BSG-structure and IC laylot applications, *Proceedings of IEEE/ACM ICCAD 1999*, pp. 484–491, 1999.
20. T. Ohtsuki, N. Suzigama, and H. Hawanishi, An optimization technique for integrated circuit layout design, *Proceedings of ICCST 1970*, pp. 67–68, 1970.
21. B. Obermeier and F. Johannes, Temperature aware global placement, *Proceedings of ASPDAC 2004*, pp. 143–148, 2004.
22. R. H. J. M. Otten, Automatic floorplan design, *Proceedings of ACM/IEEE DAC 1982*, pp. 261–267, 1982.
23. R. H. J. M Otten, Efficient floorplan optimization, *Proceedings of IEEE ICCD 1983*, pp. 499–502, 1983.
24. S. Palacharla, N. P. Jouppi, and J. E. Smith, Complexity-effective superscalar processors, *Proceedings of the 24th ISCA*, pp. 206–218, June 1997.
25. K. Puttaswamy and G. Loh, The impact of 3-dimensional integration on the design of arithmetic units, *Proceedings of ISCAS 2006*, pp. 4951–4954, May, 2006.
26. K. Puttaswamy and G. Loh, Dynamic instruction schedulers in a 3-dimensional integration technology, *Proceedings of ACM/IEEE GLS-VLSI 2006*, pp. 153–158, May 1, 2006, USA.
27. G. Reinman and N. Jouppi, Cacti 2.0: An integrated cache timing and power model, *In Technical Report*, 2000.
28. R. Ronnen, A. Mendelson, K. Lai, S. Liu, F. Pollack, and J. Shen, Coming challenges in microarchitecture and architecture, *Proceedings of the IEEE*, 89(3): 325–340, 2001.
29. Z. C. Shen and C. C. N. Chu, Bounds on the number of slicing, mosaic, and general floorplans, *IEEE Transaction on CAD*, 22(10): 1354–1361, 2003.
30. Y. Tsai, Y. Xie, N. Vijaykrishnan, and M. Irwin, Three-dimensional cache design exploration using 3D CACTI, *Proceedings of ICCD 2005*, pp. 519–524, October 2005.
31. M. Tremblay, B. Joy, and K. Shin, A three dimensional register file for superscalar processors, *Proceedings of the 28th HICSS*, pp. 191–201, 1995.
32. X. Tang, R. Tian and D. F. Wong, Fast evaluation of Sequence Pair in block placement by longest common subsequence computation, *Proceedings of DATE 2000*, pp. 106–111, 2000.
33. X. Tang and D. F.Wong, FAST-SP: A fast algorithm for block placement based on Sequence Pair, *Proceedings of ASPDAC 2001*, pp. 521–526, 2001.
34. D. F. Wong and C. L. Liu, A new algorithm for floorplan design, *Proceedings of the 3rd ACM/IEEE DAC*, pp. 101–107,1986.
35. H. Yamazaki, K. Sakanushi, S. Nakatake, and Y. Kajitani, The 3D-packing by meta data structure and packing heuristics, IEICE Transaction on Fundamentals, E82-A(4): 639–645, 2000.
36. T. Yan, Q. Dong, Y. Takashima, and Y. Kajitani, How does partitioning matter for 3D floorplanning?, *Proceedings of the 16th ACM GLS-VLSI*, pp. 73-78, 2006.
37. Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, ISAC: Integrated space and time adaptive chip-package thermal analysis, *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 26(1): 86–99, January 2007.
38. B. Yao, H. Chen, C. K. Cheng and R. Graham, Floorplan representations: complexity and connections, *ACM Transaction on Design Automation of Electronic Systems*, 8(1): 55–80, 2003.
39. E. F. Y. Young, C. C. N. Chu, and Z. C. Shen, Twin Binary Sequences: A nonredundant representation for general nonslicing floorplan, *IEEE Transaction on CAD*, 22(4): 457–469, 2003.

40. S. Zhou, S. Dong, C.-K. Cheng, and J. Gu, ECBL: An extended Corner Block List with solution space including optimum placement, *Proceedings of ISPD 2001*, pp. 150–155, 2001.
41. P. Zhou, Y. Ma, Z. Li, R. P. Dick, L. Shang, H. Zhou, X. Hong, and Q. Zhou, 3D-STAF: scalable temperature and leakage aware floorplanning for three-dimensional integrated circuits, *Proceedings of ICCAD 2007*, pp. 590–597, 2007.
42. C. Zhuang, Y. Kajitani, K. Sakanushi, and L. Jin, An enhanced Q-Sequence augmented with empty-room-insertion and parenthesis trees, *Proceedings of DATE 2002*, pp. 61–68, 2002.

第5章　热敏感三维（3D）布局

Jason Cong　Guojie Luo

3D IC 技术为电路设计提供了另一维的自由度。在处理以更优线网长度进行的单元的器件层指定优化之外，同时要处理硅通孔（TSV）资源和热问题，这对布局工具来说是一个大挑战。本章介绍了几种针对这些问题的3D全局布局技术，包括基于分块的技术、二次均匀建模技术、多层布局技术和基于转换的技术。也简要地介绍了3D IC设计的合法化问题和详细布局问题，并且通过实验结果说明了多种3D布局技术对线网长度、TSV数和温度的影响，以及3D IC工艺对线网长度和中继器使用的影响。

5.1　介绍

布局是物理设计流程中重要的一步。布局结果的质量严重影响性能、功耗、温度和可布线度。而3D IC 技术给其中的热问题带来了更多的挑战：①垂直堆叠的多层有源器件使功率密度激增；②相比于硅和金属来说，在器件层间电介质的热导率很低。例如 SiO_2 的热导率在室温（300K）下为 1.4W/m · K[28]，其大大小于硅（150W/m · K）和铜（401W/m · K）的热导率。因此，在3D IC设计的每个阶段，包括布局阶段，都要考虑热问题。所以，对于充分应用3D IC技术，热敏感的3D布局工具是必要的。读者可参看3.2节了解热问题的详细介绍和热分析及热优化的方法。

5.1.1　问题建模

给定电路 $H=(V, E)$，器件层数为 K，每层布局区域 $R=[0,a]\times[0,b]$，其中在电路 H（用超图表示）中 V 是例化单元集（用顶点来表示），E 为线网集（用超边表示），单元 $v_i\in V$ 的布局（x_i, y_i, z_i）满足（x_i, y_i）$\in R$ 且 $z_i\in\{1, 2, \cdots, K\}$。3D布局问题即为每个单元 $v_i\in V$ 找到一个布局（x_i, y_i, z_i），根据不重叠布局约束、性能约束、温度约束等约束使加权的总线网长度的目标函数最小化。在本章中我们重点关注温度约束，而性能约束与2D设计中的类似。读者可参考本章文献［18，35］做2D布局的简单了解和学习。

5.1.1.1　线网长度目标函数

布局解的质量可用性能、功耗和可布线度衡量，但这些衡量是不统一的。为了在优化过程中给这些方面建模，加权总线网长度成为被广泛接受的布局质量衡量单

位[34,35]。通常，目标函数定义为

$$\mathrm{OBJ} = \sum_{e \in E} (1 + r_e) \cdot (WL(e) + \alpha_{TSV} \cdot TSV(e)) \tag{5-1}$$

目标函数依赖布局结果$\{(x_i, y_i, z_i)\}$，并且其为所有线网长度 $WL(e)$ 和硅通孔（TSV）数 $TSV(e)$ 的加权和。权重 $(1+r_e)$ 反映了线网 e 的临界度，其通常与性能优化相关。未加权线网长度通过设 r_e 为 0 得到。通过将该权重涉及热阻、电容和线网 e 的开关行为，就能够对热效应建模[27]。

线网长度 $WL(e)$ 通常用半周长线网长度模型估算[27,19]，即

$$WL(e) = (\max_{v_i \in e}\{x_i\} - \min_{v_i \in e}\{x_i\}) + (\max_{v_i \in e}\{y_i\} - \min_{v_i \in e}\{y_i\}) \tag{5-2}$$

同理，$TSV(e)$ 由 $\{z_i: v_i \in e\}$ 建模：

$$TSV(e) = (\max_{v_i \in e}\{z_i\} - \min_{v_i \in e}\{z_i\}) \tag{5-3}$$

系数 α_{TSV}是 TSV 的权重；其将 TSV 建模为一定长度的线网。例如，0.18μm SOI（硅上绝缘体）技术[22]就电阻的角度评价一个 3μm 厚的 TSV 相当于 8～20μm 的第 2 层金属线。因此在 8～20μm 间的系数 α_{TSV}可用于优化此种情况下的功耗或延迟。

5.1.1.2 不重叠约束

不重叠约束的终极目标可以表示如下：

$$\begin{aligned} &|x_i - x_j| \geq (w_i + w_j)/2 \\ \text{或} \quad &\text{对于所有 } z_i = z_j \text{的单元 } v_i, v_j \\ &|y_i - y_j| \geq (h_i + h_j)/2 \end{aligned} \tag{5-4}$$

式中，(x_i, y_i, z_i) 是单元 i 的布局；w_i 和 h_i 分别是其宽度和高度。单元 j 与此相同。这样的约束直接应用在某些较早的解析布局工具上，如本章参考文献 [5]。

然而，该公式导致了数量为 $O(n^2)$ 二选一约束，其中 n 是单元的总数。这样数量的约束对现代大规模设计不实用。

为了建模并处理这些成对的不重叠约束，现代布局器使用了更可按比例缩小的步骤将布局分为粗合法化和精细合法化。通过使用如下的区域密度约束，粗合法化松弛了成对的不重叠约束。

$$\sum_{\substack{\text{对于所有} \\ z_i = k\text{的单元}_i}} \text{重叠}(\text{桶}_{m,n,k}, \text{单元}_i) \leq \text{面积}(\text{桶}_{m,n,k}) \quad (\text{对于所有 } m,n,k) \tag{5-5}$$

对于具有 K 个器件层的 3D 电路，每层分为 $L \times M$ 个桶。如果每个桶$_{l,m,k}$满足等式（5-5），粗合法化即完成。一个器件层密度约束的例子见图 5-1。

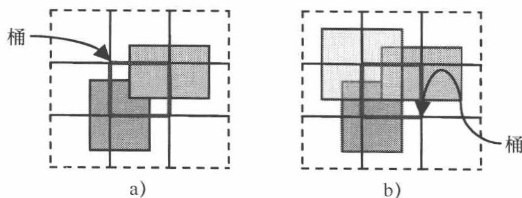


图 5-1 a）满足密度约束；b）不满足密度约束

在粗合法化之后，精细合法化用来满足成对的不重叠约束，使用

了多种分立方法和试探法，这些方法将在5.6节中说明。

5.1.1.3　热敏感

在现有文献中，温度问题没有直接建模为约束。代替的是，热惩罚加入线网长度目标函数以控制温度。该惩罚既可以是转换为热敏感线网权重的加权温度惩罚[27]，也可以是热分布成本惩罚[41]，再或者是在合法化过程中从单元位置到热沉的距离[19]。

在本章中，我们将在5.2节介绍热敏感线网权重，在5.3.3节将介绍热分布成本函数，在5.6.2.2节将介绍热敏感合法化。

5.1.2　现有三维布局技术总览

当前最先进的2D布局算法分为打平布局技术、顶到底基于分块的技术和多层布局技术[35]。这些技术展示出了现代VLSI电路不断增长的复杂度可按比例的缩小性。为了处理按比例缩小问题，这些技术将布局问题分解为3个阶段：全局布局、合法化和详细布局。给定初始解，全局布局改进解直到在每个预定义区域中的单元面积不大于该区域容积。这些区域被基于分块的技术和多层布局技术以自顶向下的方式从最粗糙的层次处理到最精细的层次，并在最精细层次以打平的形式用打平布局技术处理。在全局布局后，合法化执行以在不重叠的情况下确定所有单元的特定位置，并由详细布局执行本地精细化以得到最终解。

随着现代2D布局技术的进步，也有许多3D布局技术被开发出来以针对3D技术的问题。大多数现有技术，特别是在全局布局阶段，都可视为2D布局技术的扩展。我们将3D布局技术分为以下几类：

1）基于分块的技术[21,1,2,27]在传统基于分块的过程中一些合适的阶段将分块平面平行插入器件层间。分块的成本通过估计的线网长度和TSV数目的加权来衡量，其中线网还通过热敏感或拥塞敏感系数进一步加权来考虑温度和可布线度。

2）打平布局技术大多数是二次布局及其衍生技术，包括力定向技术、单元平移技术和二次均匀建模技术。未约束的二次布局会引入大量的单元重叠，因此开发了不同的衍生技术来消除重叠。二次方程的最小化将转化为解决线性系统的问题。力定向技术[26,33]在线性系统右手侧增加了一个向量，称为斥力向量。这些斥力向量在电荷分布与单元面积分布相同时与电场力等效。力在每次迭代后刷新直到在每个预指定区域的单元面积不大于该区域容积。单元平移技术[29]与力定向技术的相似之处在于它们也在线性系统的右手侧增加了一个向量。该向量是从虚拟引脚发出的线网力的结果，其中虚拟引脚位置是根据在单元平移后所需单元位置添加的。二次均匀建模技术[41]在目标函数上增加了一个密度惩罚函数，并且在每次迭代中通过另一个二次函数本地拟合了密度惩罚函数，因此整个全局布局可通过最小化一系列二次函数求解。

3）多层技术[13]通过初始网表构建物理结构，并解决一系列从最粗糙层次到最

精细层次的布局问题。

4）作为这些技术的补充，本章参考文献［19］中提出的3D布局方法利用了现存的2D布局结果并通过转化构建了3D布局。

在本章的后续部分，我们将详细讨论这些问题，还将特别介绍针对3D布局的合法化和细致布局技术。

5.2 基于分块的技术

基于分块的技术[21,1,3,27]可以有效地减小TSV数目及其固有的最小切割目标。这是建设性的，即使在缺少I/O压焊点连接信息时也能够得到高质量布局结果的方法。

基于分块的布局技术使用应用于3D电路的递归双向分块（对分）方法。在每一步对分时，块（V_0，R_0）由在网表中且处于布局区域R的特定物理块R_0中的单元的子集组成$V_0 \subseteq V$。当块对分时，两个新块（V_1，R_1）和（V_2，R_2）由对分的$V_0 = V_1 \cup V_2$的单元表和对分的物理区域$R_0 = R_1 \cup R_2$得到，其中切割面通常与X、Y或Z轴垂直。单元表V_0的平衡对分为$V_1 \cup V_2$通常是最优的，因为其满足了面积的平衡标准$W_i = \sum_{v \in V_i}$面积（v），其中$i = 1$，2从而使$|W_1 - W_2| \leqslant \tau(W_1 + W_2)$，$\tau$为容差。$R_1$和$R_2$的面积比率与$V_1$和$V_2$间的单元面积比率相关。在一定数量的对分步骤后，在5.1.1.2节中定义的区域密度约束由于对分过程的特性自动满足。

基于分块技术的布局解由对分目标函数和对分方向的选择决定，如下所述。

基于最小切割的布局的中心思想是最小化块间的切割面积，从而使具有较多连接关系的单元倾向于处在同一分块中并相互靠近以获得更短的线网长度。

对于（V_0，R_0）对分为（$V_1, R_1) \cup (V_2, R_2$）的情况，如果一个线网同时含R_1和R_2中的单元，那么它将被切割。总的加权切割量为$\sum_{e被切割}(1 + r_e)$。切割过程中的目标是最小化总的加权切割量，其可使用采用多层hMetis[32]方式的Fiduccia-Mattheyses（FM）试探算法[24]求解。

终端传播[23]是一种考虑块的外部连接的成功技术。在块外的一个单元建模为该块边缘上的固定终端，其位置通过线网中心的最近位置计算得到。

然而，切割量函数不直接反映在5.1.1.1节中定义的3D布局问题的线网长度目标函数，其中切割量未考虑α_{TSV}的权重。当切割面与x轴或y轴垂直时，切割量的最小化只对2D线网长度$\sum_{e \in E}(1 + \mathrm{r}_e)\mathrm{WL}(e)$有潜在影响；当切割面与$z$轴垂直时，切割量等于$\sum_{e被切割}(1 + r_e)\alpha_{TSV}TSV(e)$。折中这两个目标的唯一方法是控制对分方向的顺序。文献［21］中的研究注意到总线网长度和TSV数的折中可以通过改变何时把电路分块到器件层的顺序实现。直观地，在z轴尺度分块首先将最小化TSV数目，而在x轴和y轴尺度分块将最小化总线网长度。本章参考文献［21，27］使用权重系数α_{TSV}来决定对分方向。假定物理区域R，每次对分的切割方向选

为与区域中的最大的宽度边 $|x_U - x_L|$，高度 $|y_U - y_L|$，或加权深度 $\alpha_{TSV}|z_U - z_L|$ 垂直。由此，以在成本较低的正交方向上允许更高连接度的成本，所述最小切割目标最小化了成本最高方向上的连接数目。

式（5-6）示出了附加到未加权线网长度目标函数上的热敏感项[27]。我们将展示该函数可以被加权总线网长度代替。

$$\sum_{e \in E}(WL(e) + \alpha_{TSV} \cdot TSV(e)) + \alpha_{TEMP}\sum_{v_i \in V} T \tag{5-6}$$

式中，T_j 是单元 $cell_j$ 的温度，同时在分块过程中考虑了温度敏感度 $\alpha_{TEMP}\sum_{v_i \in V} T_i$。然而，直接在目标函数中使用温度项会导致对每个单元移动的重计算。因此，必须对此简化以提高效率。从单元 cell v_i 到周围环境的总热阻可计算得到：

$$R_i = (R_{left,i}^{-1} + R_{right,i}^{-1} + R_{front,i}^{-1} + R_{rear,i}^{-1} + R_{bottom,i}^{-1} + R_{top,i}^{-1})^{-1} \tag{5-7}$$

式中，$R_{left,i}$，$R_{right,i}$，$R_{front,i}$，$R_{rear,i}$，$R_{bottom,i}$，$R_{top,i}$ 是通过仅考虑该方向热传导情况下的有限元（差分）法（FDM，3.2.2.1 节）分析得到的近似热阻。例如，$R_{left,i}$ 作为从单元位置（x_i，y_i，z_i）到 3D 芯片左侧边缘（$x=0$）的热阻计算，单元纵切面积等于单元宽度与单元厚度的乘积。

于是实际使用的目标为

$$\begin{aligned}&\sum_{e \in E}(WL(e) + \alpha_{TSV} \cdot TSV(e)) + \alpha_{TEMP}\sum_{v_i \in V}\Delta T_i \\ =& \sum_{e \in E}(WL(e) + \alpha_{TSV} \cdot TSV(e)) + \alpha_{TEMP}\sum_{v_i \in V} R_i P_i\end{aligned} \tag{5-8}$$

式中，ΔT_j 是 v_i 的温度分布并且是 T_j 的主要部分；R_i 是从 v_i 到周围环境的热阻；P_i 是 v_i 的功率分布。为了实现热敏感度，需对 P_i 和 R_i 进行优化。

与线网 e 相关的动态功耗为

$$P_e = 0.5 a_e f V_{DD}^2 (C_{perWL} WL(e) + C_{perTSV} TSV(e) + C_{per\,pin} n_e^{\text{input pins}}) \tag{5-9}$$

式中，α_e 是活动系数；f 是始终频率；V_{DD} 是供电电压；C_{perWL} 是单位线长的电容；C_{perTSV} 是每个 TSV 的电容；$C_{per\,pin}$ 是每个输入引脚的电容；$n_e^{\text{input pins}}$ 是线网 e 驱动的单元输入引脚数目。因为单元的内部电阻通常大于线网电阻[27]，功耗 P_e 在驱动单元 i 消耗并成为 P_i 的组成部分。这些功耗组分的总和即为单元 v_i 的总功耗：

$$P_i = \sum_{\substack{\text{由}v_i\text{驱动的}\\\text{线网}e}} P_e = \sum_{\substack{\text{由}v_i\text{驱动的}\\\text{线网}e}} 0.5 a_e f V_{DD}^2 (C_{perWL} WL(e) + C_{perTSV} TSV(e) + C_{per\,pin} n_e^{\text{input pins}}) \tag{5-10}$$

如果略去在优化过程中为常数的 $C_{per\,pin}\ n_e^{\text{input pins}}$ 并用 $C_{perWL}\alpha_{TSV}$ 替代 C_{perTSV}，其中 α_{TSV} 是在 5.1.1.1 节中定义的，那么式（5-8）可表示为

$$\sum_{e \in E}(WL(e) + \alpha_{TSV} \cdot TSV(e)) + \alpha_{TEMP}\sum_{v_i \in V} R_i P_i$$

$$= \sum_{e \in E} (WL(e) + \alpha_{TSV} \cdot TSV(e)) + \alpha_{TEMP} \sum_{v_i \in V} R_i$$

$$\cdot \sum_{\text{由}v_i\text{驱动的线网}e} 0.5 a_e f V_{DD}^2 C_{perWL} (WL(e) + \alpha_{TSV} \cdot TSV(e))$$

$$= \sum_{e \in E} (WL(e) + \alpha_{TSV} \cdot TSV(e)) + \alpha_{TEMP} \sum_{e \in E}$$

$$\sum_{v_i\text{驱动线网}e} R_i \cdot 0.5 a_e f V_{DD}^2 C_{perWL} (WL(e) + \alpha_{TSV} \cdot TSV(e))$$

$$= \sum_{e \in E} (1 + \alpha_{TEMP} \sum_{v_i\text{驱动线网}e} R_i \cdot 0.5 a_e f V_{DD}^2 C_{perWL}) (WL(e) + \alpha_{TSV} \cdot TSV(e)) \tag{5-11}$$

与式（5-1）中定义的总体加权线网长度相比较，这些热敏感线网权重可定义实现如下

$$r_e = \alpha_{TEMP} \sum_{v_i\text{驱动线网}e} R_i \cdot 0.5 a_e f V_{\mathrm{DD}}^2 C_{perWL} \tag{5-12}$$

热敏感线网权重 r_e 在分块过程中不是常数。相反，热阻 R_i 由单元 v_i 与芯片边缘的距离决定。本章参考文献［27］中的简单计算可通过以下假定实现。即假定热流从单元位置沿所有3个方向直线流向芯片边缘，同时整体热阻由这些方向各自分别的热阻计算得到。这些热阻在分块过程中求值以计算将一个单元从一块移动到另一块的增益。

作为热敏感线网加权目标函数的附带结果，温度也通过将单元拉到热沉的伪线网得到优化[27]。

5.3 二次均匀建模技术

与离散基于分块的技术不同，二次基于布局的技术是连续的。其中心思想是通过较弱的约束来松弛器件层的单元 $z \in \{1, \cdots, K\}$ 的指定，其中 $z \in [1, K]$。3D布局问题通过最小化二次成本函数或找到被推导出的线性系统的解来求解。区域密度约束的处理方式有两种：一是通过在线性系统上附加一个力向量（力定向技术[26,33]和单元平移技术[29]）；二是在二次成本函数中加入二次惩罚（二次均匀建模技术[41]）。3D全局布局通过最小化一系列二次成本函数求解。在本节中，我们将讨论二次均匀建模技术。

二次的布局流程如图5-2所示。该流程分为全局布局和详细布局阶段，其中全局布局通过二次均匀建模技术求解，而详细布局通过简单的分层2D布局或其他在5.6节中将讨论的高级合法化和详细布局技术求解。

统一的二次成本函数定义为

$$OBJ^+ = OBJ + \beta \times DIST + \gamma \times TDIST \tag{5-13}$$

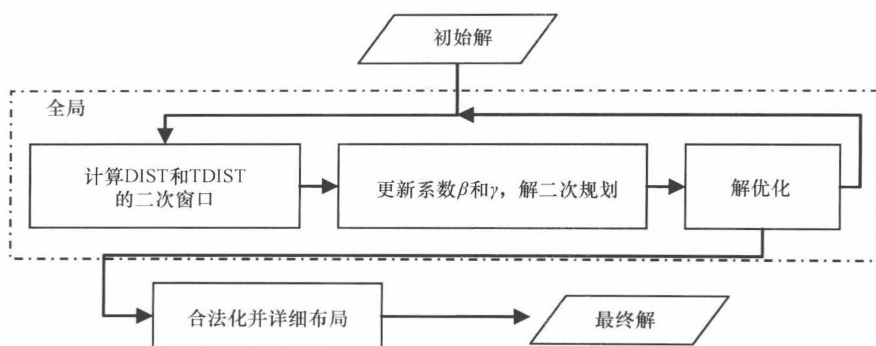


图 5-2　二次布局流程图

式中，*OBJ* 是线网长度目标在 5.1.1.1 节中的定义；DIST 是单元排布成本；β 是单元排布成本权重；*TDIST* 是热分布成本；γ 是热分布成本权重。此外，所有这些函数 *OBJ*、*DIST* 和 *TDIST* 均在式（5-14）中表示为二次形式，并将在后续章节详解。

$$
\begin{aligned}
OBJ &= \sum_{i=1}^{n}\left(\sum_{j=1}^{n} q_{x,ij}x_i x_j + p_{x,i}x_i\right) + \sum_{i=1}^{n}\left(\sum_{j=1}^{n} q_{y,ij}y_i y_j + p_{y,i}y_i\right) \\
&\quad + \sum_{i=1}^{n}\left(\sum_{j=1}^{n} q_{z,ij}z_i z_j + p_{z,i}z_i\right) + r \\
DIST &\approx \sum_{i=1}^{n}(a_{x,i}x_i^2 + b_{x,i}x_i) + \sum_{i=1}^{n}(a_{y,i}y_i^2 + b_{y,i}y_i) + \sum_{i=1}^{n}(a_{z,i}z_i^2 + b_{z,i}z_i) + C \\
TDIST &\approx \sum_{i=1}^{n}(a_{x,i}^{(T)}x_i^2 + b_{x,i}^{(T)}x_i) + \sum_{i=1}^{n}(a_{y,i}^{(T)}y_i^2 + b_{y,i}^{(T)}y_i) + \sum_{i=1}^{n}(a_{z,i}^{(T)}z_i^2 + b_{z,i}^{(T)}z_i) + C^{(T)}
\end{aligned}
\tag{5-14}
$$

5.3.1　线网长度目标函数

为了构建二次线网长度函数来估计在 5.1.1.1 节中定义的线网长度目标，多引脚线网将星形模型或集团模型分解为两引脚线网。在结果图表中，二次线网长度定义为

$$
OBJ = \sum_{\substack{e\in E \\ v_i,v_j\in e}} (1+r_e)((s_{e,x}(x_i - x_j)^2 + s_{e,y}(y_i - y_j)^2) + \alpha_{TSV}s_{e,z}(z_i - z_j)^2)
\tag{5-15}
$$

式中，$(1+r_e)$ 是线网权重；α_{TSV}是在 5.1.1.1 节中定义的 TSV 系数；线网 e 是分解得到的连接 v_i 的 (x_i, y_i, z_i) 点和 v_j 的 (x_j, y_j, z_j) 点的两引脚线网；系数 $s_{e,x}$、$s_{e,y}$、$s_{e,z}$能线性化二次线网长度以估计在公式（5-2）和（5-3）中定义的 HPWL 线网长度和 TSV 数目[38]。

很明显这种二次函数 OBJ 能写成矩阵形式：

$$OBJ = \sum_{i=1}^{n}\left(\sum_{j=1}^{n} q_{x,ij}x_i x_j + p_{x,i}x_i\right) + \sum_{i=1}^{n}\left(\sum_{j=1}^{n} q_{y,ij}y_i y_j + p_{y,i}y_i\right) + \sum_{i=1}^{n}\left(\sum_{j=1}^{n} q_{z,ij}z_i z_j + p_{z,i}z_i\right) + r \tag{5-16}$$

式中，x_i、y_i、z_i 是问题变量；系数 $q_{x,ij}$、$p_{x,i}$、$q_{y,ij}$、$p_{y,i}$、$q_{z,ij}$、$p_{z,i}$和 r 可直接从公式（5-15）计算得到。系数 $p_{x,i}$、$p_{y,i}$、$p_{z,i}$和 r 与电路中 I/O 引脚和固定单元的位置有关。

5.3.2　单元排布成本函数

在 2D 布局中使用离散余弦变换来求单元排布并帮助展开单元的初始思想来源于本章参考文献［42］。该思想扩展并应用到了 3D 布局中。

与在 5.1.1.2 中定义的桶密度相似，另外一种用于含连续变量（z_i）的松弛问题的桶密度定义为

$$d_{m,n,l} = \frac{\sum_{\text{对于所有单元}i} \text{交集}(\text{桶}_{m,n,l}, \text{单元 } i)}{\text{体积}(\text{桶}_{m,n,l})} \tag{5-17}$$

假定一个具有 K 个器件层的 3D 电路，其芯片宽度为 W，高度为 H，松弛布局区域$[0,W]\times[0,H]\times[0,K]$ 分为 $M\times N\times L$ 个桶，其中在（x_i，y_i，z_i）的单元 cell_i 映射到区域$[x_i - w_i/2, x_i + w_i/2]\times[y_i - n_i/2, y_i + n_i/2]\times[z_i, z_i+1]$。

$\{f_{p,q,v}\} = DCT(\{d_{m,n,l}\})$ 的 3D DCT 转换定义为

$$f_{p,q,v} = \sqrt{\frac{8}{MNL}}C(p)C(q)C(v)\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{l=0}^{L-1} d_{m,n,l}\cos\left(\frac{(2m+1)p\pi}{2M}\right)\cos\left(\frac{(2n+1)q\pi}{2N}\right)\cos\left(\frac{(2l+1)v\pi}{2L}\right) \tag{5-18}$$

式中，m、n、l 是空间区域的坐标，p、q、v 是频域的坐标。系数为 $C(v) = \begin{cases} 1/\sqrt{2} & t=0 \\ 1 & \text{其他} \end{cases}$。

单元分布成本定义为

$$DIST = \sum_{p,q,t} u_{p,q,t} f_{p,q,t}^2 \tag{5-19}$$

式中，$u_{p,q,t} = 1/(p+q+t+1)$ 被试探性地设置。

注意到式（5-19）不是与布局变量（x_i，y_i，z_i）相关的二次函数。为了构建二次形式，进行了如下近似：

$$DIST \approx \sum_{i=1}^{n}(a_{x,i}x_i^2 + b_{x,i}x_i) + \sum_{i=1}^{n}(a_{y,i}y_i^2 + b_{y,i}y_i) + \sum_{i=1}^{n}(a_{z,i}z_i^2 + b_{z,i}z_i) + C \tag{5-20}$$

虽然系数 $a_{x,i}$、$b_{x,i}$、$a_{y,i}$、$b_{y,i}$、$a_{z,i}$、$b_{z,i}$依赖中间布局结果，他们仍假定为该二

次函数的常数。这些系数在中间布局改变时更新。由于变量在该近似过程中可以较好地消去，系数可依次算出。为了计算 $a_{x,i}$、$b_{x,i}$，所有除 x_i 以外的变量均固定，由此成本函数变为 x_i 的二次函数：

$$DIST(x_i) \approx a_{x,i}x_i^2 + b_{x,i}x_i + C'_{i,x} \tag{5-21}$$

3 个系数 $a_{x,i}$、$b_{x,i}$ 和 $C'_{i,x}$ 由 3 个成本函数 $DIST(x_i)$、$DIST(x_i+\delta)$ 和 $DIST(x_i\delta)$ 算出。通过计算，我们可以看到满足二次近似的一阶和二阶导数导致该二次函数的一阶和二阶导数分别局部近似了区域分布成本函数 $DIST$ 的一阶和二阶导数。

$$2a_{x,i}x_i + b_{x,i} = \frac{DIST(x_i+\delta) - DIST(x_i-\delta)}{2\delta} \approx \frac{\partial DIST(x_i)}{\partial x_i}$$

$$2a_{x,i} = \frac{DIST(x_i+\delta) - 2DIST(x_i) + DIST(x_i-\delta)}{\delta^2} \approx \frac{\partial^2 DIST(x_i)}{\partial^2 x_i} \tag{5-22}$$

多重 $DIST$ 函数的计算避免了通过预计算的 3D DCT 变换[42]。其在式（5-20）中计算矩阵系数的过程中以 $O(n)$ 的运行时间消耗了 $O(M^2N^2L^2)$ 的空间。

5.3.3　热分布成本函数

热成本的处理类似于单元分布成本，通过将单元密度 $\{d_{m,n,l}\}$ 替换为热密度 $\{t_{m,n,l}\}$。热密度定义为

$$t_{m,n,l} = T_{m,n,l}/T_{avg} \tag{5-23}$$

式中，$T_{m,n,l}$是桶 m，n，l 中的平均温度；T_{avg}是整个芯片的平均温度。

作为单元分布的成本，热分布通过 3D DCT 转换，分布成本函数通过二次形式近似。

在热分布函数 TDIST 的二次近似过程中，除了矩阵系数的计算，热密度 $\{t_{m,n,l}\}$ 的计算也消耗了大量运行时间，原因是准确地计算需要热分析。为了从 $DIST(x_i)$，$DIST(x_i-\delta)$ 和 $DIST(x_i+\delta)$ 的计算过程中的热分析过程节省时间，在计算新 $\{t_{m,n,l}\}$ 时做了近似。本章参考文献［41］中的工作使用了两种近似方法，可能都缺乏准确度但速度快，可集成到分布成本计算中。

第一种近似利用了单元的热分布。令 $P_{bin}(i)$ 和 $T_{bin}(i)$ 作为桶 $m(i)n(i)l(i)$ 的功率和平均温度，在此桶中单元的热分布定义为

$$T_{cell} = \frac{P_{cell}}{P_{bin}(i)}T_{bin}(i) \tag{5-24}$$

当单元从桶 $m(i)n(i)l(i)$ 移动到桶 $m(j)n(j)l(j)$ 时，桶的温度更新为

$$T_{bin}(i) \leftarrow T_{bin}(i) - \beta \cdot T_{cell}$$
$$T_{bin}(j) \leftarrow T_{bin}(j) - \beta \cdot T_{cell} \tag{5-25}$$

式中，$\beta = \mathrm{l}(j)/\mathrm{l}(i)$ 是单元对桶温度的影响。

第二个近似以与更新功率密度相同的比率更新了桶温度：

$$T'_{bin}(i) = \frac{P'_{bin}(i)}{P_{bin}(i)} T_{bin}(i) \tag{5-26}$$

5.4 多层布局技术

多层试探法[15]已证明在大规模设计中有效。多层试探法在分块问题中的应用[32]也显示了其可以改善解的质量；这也涉及5.2节中讨论的基于分块的技术。此外，二次基于布局问题的求解器通常使用多格点方法，此方法是多层试探法的起源方法。

在本节中，我们将介绍直接利用多层试探法的解析3D布局引擎。

5.4.1 三维布局流程

多层解析3D布局流程如图5-3所示。全局布局从随意尝试或引入给定初始布局开始。全局布局将解析布局引擎（5.4.2节）合并入本章参考文献［15］中使用的多层框架。全局布局随后用2D详细布局器[16]分层进行以获取最终布局。

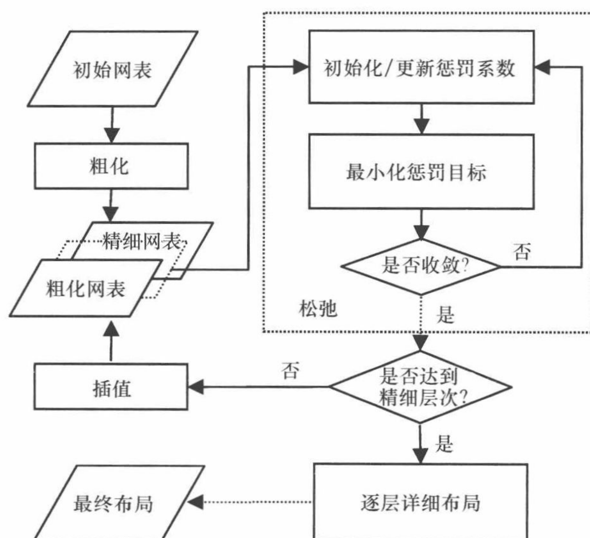


图5-3 多层解析3D布局流程

5.4.2 解析布局引擎

对于多层试探法，解析布局不是唯一的引擎。实际上，任何与5.3节中介绍的技术类似的打平3D布局技术均可使用。在本节中，我们重点关注最早将多层试探法应用于3D布局的解析引擎[13]。

解析布局引擎通过将非重叠约束转换为密度惩罚来解决3D全局布局问题。

$$\text{最小化} \sum_{e \in E} (WL(e) + \alpha_{TSV} \cdot TSV(e)) \\ \text{条件为 } Penalty(\vec{x}, \vec{y}, \vec{z}) = 0 \tag{5-27}$$

线网长度 $WL(e)$（5.4.2.2 节），TSV 数目 $TSV(e)$（5.4.2.3 节）和密度惩罚函数 $Penalty(\vec{x}, \vec{y}, \vec{z})$（5.4.2.4 节）将在以下章节详细介绍。

为了解该受约束问题，惩罚方式[37]通常使用：

$$OBJ(\vec{x}, \vec{y}, \vec{z}) = \sum_{e \in E} (WL(e) + \alpha_{TSV} \cdot TSV(e)) + \mu \cdot P(\vec{x}, \vec{y}, \vec{z}) \tag{5-28}$$

每次迭代，逐步增加惩罚系数 μ 来避免密度违反，使该惩罚目标函数最小化。当 $\mu \to \infty$ 时，若惩罚函数非负，则式（5-28）的最小值与式（5-27）等价。

5.4.2.1　离散变量的松弛化

如 5.1.1 节中提到的，布局变量由三元变量（x_i，y_i，z_i）表示，其中 z_i 是离散变量 {1, 2, …, K}。z_i 的范围从集合 {1, 2, …, K} 松弛到连续区间 [1, K]。在松弛化后，就可在我们的布局引擎中使用非线性解析器。在详细布局过程前松弛解映射回离散值。

5.4.2.2　对数—和—指数线网长度

式（5-2）中定义的半周长线网模型 $WL(e)$ 由对数—和—指数函数[4]的可微近似替代，其引入本章参考文献［36］中的布局。

$$WL(e) \approx \eta\Big(\log \sum_{v_i \in e} \exp(x_i/\eta) + \log \sum_{v_i \in e} \exp(-x_i/\eta) + \\ \log \sum_{v_i \in e} \exp(y_i/\eta) + \log \sum_{v_i \in e} \exp(-y_i/\eta)\Big) \tag{5-29}$$

为了数字稳定性，布局区域 R 缩小到 $[0,1] \times [0,1]$ 的范围内，因此变量（x_i，y_i）的范围也在 0 ~ 1 之间，同时参数 η，如本章参考文献［6］中设为 0.01。

5.4.2.3　TSV 数目

在式（5-3）中定义的 TSV 数目 $TSV(e)$ 的估计值也被对数—和—指数近似替换：

$$TSV(e) \approx \eta\Big(\log \sum_{v_i \in e} \exp(z_i/\eta) + \log \sum_{v_i \in e} \exp(-z_i/\eta)\Big) \tag{5-30}$$

5.4.2.4　密度惩罚函数

密度惩罚函数用于在（x，y）方向和 z 方向上去除重叠。密度惩罚函数的最小化将实现理论上的非重叠布局。

假定每个单元 v_i 具有合法器件层指定（例如，$z_i \in$ {1, 2, …, K}），那么就可为这 K 个器件层定义 K 个密度函数。直观地，密度函数 $D_k(u, v)$ 可以表示出在第 k 器件层覆盖了点（u，v）的单元数。可以定义为

$$D_k(u,v) = \sum_{i: z_i = k} d_i(u,v) \tag{5-31}$$

即为在点（u，v）上指定到该器件层的单元 v_i 的密度贡献 $d_i(u, v)$ 之和。密度贡

献 $d_i(u, v)$ 为 1 表示 v_i 在该区域中，0 表示 v_i 在该区域外。图 5-4 的例子示出两个重叠单元的密度函数。

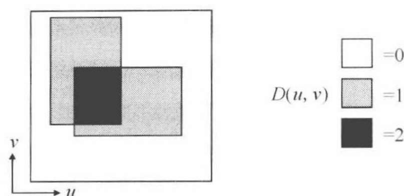


图 5-4　密度函数示例

在全局布局过程中，单元 v_i 可能停留在两器件层之间，将导致变量 $z_i \in [1, K]$ 无法对齐到两个器件层的任何一层。我们借用本章参考文献［31］的钟形函数的思想来定义这种情况下的密度函数：

$$D_k(u,v) = \sum_i \eta(k,z_i) d_i(u,v), \quad 1 \leqslant k \leqslant K \tag{5-32}$$

式中

$$\eta(k,z) = \begin{cases} 1-2(z-k)^2 & |z-k| \leqslant 1/2 \\ 2(|z-k|-1)^2 & 1/2 < |z-k| \leqslant 1 \\ 0 & \text{其他} \end{cases} \tag{5-33}$$

我们称式（5-33）为钟形密度投射函数，其将式（5-31）的密度函数中整型的层指定扩展到用于松弛？很明显式（5-32）与式（5-31）在层指定 $\{z_i\}$ 为整数时是一致的。

对于所述扩展如何在一个 4 层的 3D 布局上实现的例子在图 5-5 中给出。x 轴是 z 方向松弛化的层指定，而 y 轴表示将投射到实际器件层中的面积量。4 种曲线，短线—点曲线、点曲线、粗线曲线和短线曲线分别代表对应 1、2、3、4 器件层的函数 $\eta(1, z)$、$\eta(2, z)$、$\eta(3, z)$、$\eta(4, z)$。在本例中，单元现置于第 2 层和第 3 层之间的 $z=2.316$（x 轴上的三角）处。钟形密度投射函数将其 80% 的区域投射到第 2 层（y 轴上的偏上的三角）和 20% 区域投射到第 3 层的（Y 轴上的偏下的三角）。通过这种方式，我们建立了从松弛 3D 布局到离散层区域分布的映射关系。

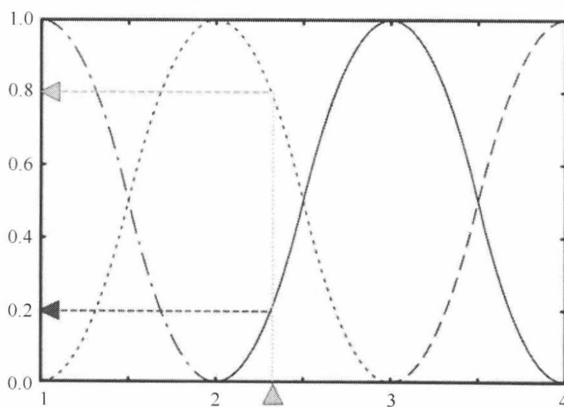


图 5-5　钟形密度投影示例

受 2D 布局方法[6,31,9]中二次惩罚项的启发，我们定义以下密度惩罚函数来测量重叠的数量：

$$P(\vec{x},\vec{y},\vec{z}) = \sum_{k=1}^{K}\int_0^1\int_0^1 (D_k(u,v) - 1)^2\mathrm{d}u\mathrm{d}v \tag{5-34}$$

定理 1　假定单元的总面积等于布局面积（例如 $\sum_i \mathrm{area}(v_i)$，没有空闲空间），每个满足对于不带有任何非整数 z_i^* 的每个 k 和 (u, v) 均满足 $D_k(u, v)=1$ 的合法布局（$\vec{x}^*$，$\vec{y}^*$，$\vec{z}^*$），是 $P(\vec{x}, \vec{y}, \vec{z})$ 的最小值。

定理 1 的证明很简单，故省略。因此，最小化 $P(\vec{x}, \vec{y}, \vec{z})$ 提供了合法布局的必要条件。由此，存在不能构建合法布局的最小值。图 5-6 示出的例子中布局 b 也最小化了密度惩罚函数，但结果不合法。

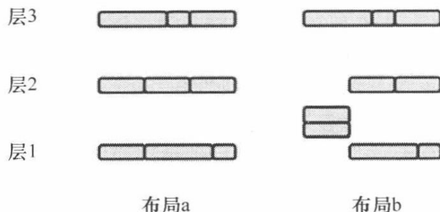


图 5-6　具有相同密度惩罚的布局

为了避免得到这样的最小值，我们引入了中间层密度函数：

$$E_k(u,v) = \sum_i \eta(k+0.5,z_i)d_i(u,v) \quad 1 \leqslant k \leqslant K-1 \tag{5-35}$$

和中间层密度惩罚函数：

$$Q(\vec{x},\vec{y},\vec{z}) = \sum_{k=1}^{K-1}\int_0^1\int_0^1 (E_k(u,v) - 1)^2\mathrm{d}u\mathrm{d}v \tag{5-36}$$

与密度惩罚函数 $P(\vec{x}, \vec{y}, \vec{z})$ 类似，以下的定理 2 也为真。

定理 2　假定单元总面积等于布局面积，每个合法布局是 $Q(\vec{x}, \vec{y}, \vec{z})$ 的最小值。

将密度惩罚函数 $P(\vec{x}, \vec{y}, \vec{z})$ 和 $Q(\vec{x}, \vec{y}, \vec{z})$ 组合，我们定义了以下的密度惩罚函数：

$$Penalty(\vec{x},\vec{y},\vec{z}) = P(\vec{x},\vec{y},\vec{z}) + Q(\vec{x},\vec{y},\vec{z}) \tag{5-37}$$

定理 1 假设单元总面积等于布局面积，那么每个合法布局（$\vec{x}^*$，$\vec{y}^*$，$\vec{z}^*$）是 $Penalty(\vec{x}, \vec{y}, \vec{z})$ 的最小值，反之亦然。

证明：很明显，通过组合定理 1 和定理 2 可知，每个合法布局是 $\mathrm{Penalty}(\vec{x}, \vec{y}, \vec{z})$ 的最小值。我们应该证明每个 $\mathrm{Penalty}(\vec{x}, \vec{y}, \vec{z})$ 的最小值（$\vec{x}^*$，$\vec{y}^*$，$\vec{z}^*$）是合法布局。从定理 1 和定理 2 的证明，我们知道当且仅当每个 k 和 (u, v) 的 $D_k(u, v)=1$ 且 $E_k(u, v)=1$ 时将得到 $\mathrm{Penalty}(\vec{x}, \vec{y}, \vec{z})$ 的最小值。首先，如果所有 $\vec{z}^*$ 的元素为整数，容易发现布局合法。其原因是所有单元指定到某个器件层，且对于任意器件层 k 上的任意点 (u, v) 仅有一个单元覆盖该点（无重叠）。

之后，我们示出不存在非整数值的 z_i^*（反之可证）。如果单元 v_i 具有非整数

z_i^*，我们知道因为$\sum_{k=1}^{K} D_k(x_i^*, y_i^*) = K$，所以有 K 个单元覆盖（x_i^*，y_i^*）。根据鸽子洞原理，因为所有变量 $\{z_i^*\}$ 在 $[1, K]$ 的范围内，这 K 个单元中至少有两个单元 v_{i1}，v_{i2} 在 z 轴的距离 $|z_{i1}^* - z_{i2}^*| < 1$。在不损失一般性的情况下，我们假定 $z_{i1}^* \leqslant z_{i2}^*$，那么将存在整数 $k \in \{1, 2, \cdots, K\}$ 使 $z_{i1}^* \in (k, k+0.5)]$ 且 $z_{i2}^* \in (k, k+1.5)$，或 $z_{i1}^* \in (k-0.5, k)$ 且 $z_{i2}^* \in (k-0.5, k+1)$ 两者之一成立。容易验证在前者情况下 $|z_{i1}^* - (k+0.5)| + |z_{i2}^* - (k+0.5)| < 1$ 且 $E_k(x_i^*, y_i^*) \geqslant \eta(k+0.5, z_{i1}^*) + \eta(k+0.5, z_{i2}^*) > 1$。在后面这种情况下，$|z_{i1}^* - k| + |z_{i2}^* - k| < 1$，$|D_k(x_i^*, y_i^*)| \geqslant y(k, z_{i1}^*) + y(k, z_{i2}^*) > 1$。这两种情况都导致 $E_k(x_i^*, y_i^*) > 1$ 或 $D_k(x_i^*, y_i^*) > 1$，其与（$\vec{x}^*$，$\vec{y}^*$，$\vec{z}^*$）是 Penalty($\vec{x}$，$\vec{y}$，$\vec{z}$）最小值的假定冲突。

因此，只存在一个非整数 z_i^*，并且 Penalty($\vec{x}$，$\vec{y}$，$\vec{z}$）的每个最小值均是 z 方向的合法布局。

在解析布局引擎中，密度 $D_k(u, v)$ 和 $E_k(u, v)$ 用平滑密度 $\widehat{D}_k(u, v)$ 和 $\widehat{E}_k(u, v)$替代以示区别。在本章参考文献［6］中，密度通过解赫尔姆霍茨（Helmholtz）等式平滑化，即

$$\begin{aligned}\widehat{D}_k(u,v) &= -\left(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial u^2} - \varepsilon\right)^{-1} D_k(u,v) \\ \widehat{E}_k(u,v) &= -\left(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial u^2} - \varepsilon\right)^{-1} E_k(u,v)\end{aligned} \tag{5-38}$$

同时在我们的实现的过程中使用了平滑化的密度惩罚函数，即

$$\begin{aligned}Penalty(\vec{x},\vec{y},\vec{z}) = &\sum_{k=1}^{K}\int_0^1\int_0^1(\widehat{D}_k(u,v) - 1)^2\mathrm{d}u\mathrm{d}v + \\ &\sum_{k=1}^{K-1}\int_0^1\int_0^1(\widehat{E}_k(u,v) - 1)^2\mathrm{d}u\mathrm{d}v\end{aligned} \tag{5-39}$$

其梯度可以用本章参考文献［12］中的方法高效计算。

5.4.3　多层架构

以下的优化问题总结了我们的解析布局引擎，即

$$\begin{cases}\text{最小化} & \begin{aligned}&\sum_{e\in E}(WL(e) + \alpha_{TSV}\cdot TSV(e)) + \\ &\mu\Big(\sum_{k=1}^{K}\int_0^1\int_0^1(\widehat{D}_k(u,v) - 1)^2\mathrm{d}u\mathrm{d}v + \\ &\sum_{k=1}^{K-1}\int_0^1\int_0^1(\widehat{E}_k(u,v) - 1)^2\mathrm{d}u\mathrm{d}v\Big)\end{aligned} \\ & \text{增加 } \mu \text{ 直到密度惩罚足够小}\end{cases} \tag{5-40}$$

该解析引擎与本章参考文献［15］中的含粗化、松弛和插值的多层架构合并。

粗化的目的是构建多层表结构，其中我们使用最优超图簇[2]。

在结构设置完毕后，多重布局问题通过最粗层次到最精细层次求解。在较粗层次，簇建模为单元，簇间的互连建模为线网，这样每个层次都有了一个布局问题。每层次的布局问题通过解析引擎式（5-40）求解（松弛）。

这些布局问题从最粗糙层次到最精细层次的顺序求解，其中在较粗层次的解做插值以获得下一个较细致层次的初始解。簇中拥有最高级别的单元置于该簇的中心（C 点），而其他单元置于与它们邻近的 C 点的加权平均位置，其权重与到这些簇的连接性成正比。

5.5 基于变换的技术

基于变换的技术[19]的中心思想是将现存 3D 布局结果通过两步过程生成 3D 热敏感布局，两个步骤为通过层指定的 3D 变换和精细化。在本节中，我们将介绍 3D 变换，包括本地堆叠变换、基于折叠的变换以及基于窗口的堆叠/折叠变换。通过层再指定的精细化与所有技术相同，将在 5.6.3 节中介绍。

基于变换的 3D 布局技术的架构如图 5-7 所示。带短线边缘的部分是基于变换的技术利用的现有的 2D 布局工具。2D 线网长度驱动和/或热驱动布局器首先用于生成目标设计的 2D 布局，即在布局区域面积等于总的 3D 布局面积。最终 3D 布局的质量高度依赖于该初始布局。随后根据给定的 3D 工艺，该 2D 布局转换为合法 3D 布局。在变换中考虑了线网长度、TSV 数目和温度。通过层重指定的精细化过程将在 3D 转换后进行以进一步减少 TSV 数并降低片上最高温度。最后，2D 详细布局器将对每一个器件层进一步精细化布局结果。

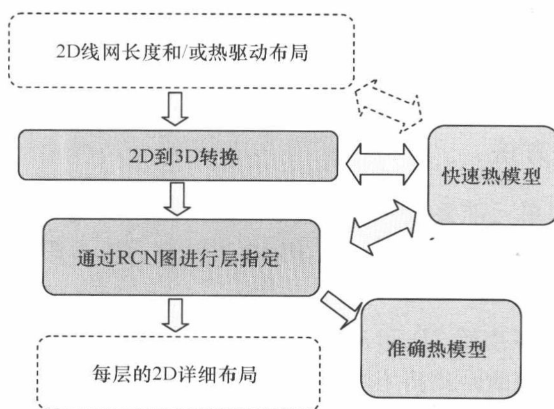


图 5-7 基于转换技术的框图

基于转换的技术始于面积 K 倍于 3D 芯片一个器件层面积的 2D 布局，其中 K 是器件层的数目。给定优化了线网长度的 2D 布局解，我们可以执行本地堆叠转换

以实现同一电路在 3D IC 工艺下更短的线网长度。我们也可以应用基于折叠的转换方式，2 折叠或 4 折叠，其能够以极低的 TSV 数目生成 3D 布局。此外，TSV 数目和线网长度的折中能通过基于窗口堆叠/折叠的方法实现。所有这些转换方法能保证在初始 2D 初始布局基础上的线网长度缩减。

5.5.1　本地堆叠转换方法

本地堆叠转换（LST）由两步组成，堆叠和合法化，如图 5-8 所示。堆叠步骤均匀地缩小芯片但不能缩小单元面积从而使单元堆叠在一个原区域 $1/K$ 面积的区域中并保持初始的相对位置。合法化步骤通过单元的位置指定最小化最大片上温度和 TSV 数目。LST 的结果是合法的 3D 布局。

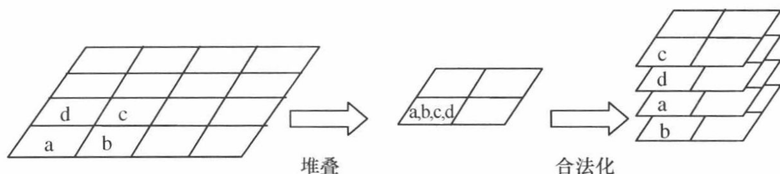


图 5-8　本地堆叠转换

对于 K 个器件层的设计，如果初始 2D 布局面积为 S，那么每层 3D 单元的面积为 S/K。在堆叠步骤中，初始布局的宽度和长度以 $K^{1/2}$ 的比率缩小，以保持初始芯片的宽高比。单元 i 的位置（x_i，y_i）也转换到新的位置（x_i'，y_i'），其中 $x_i' = x_i/k^{1/2}$，$y_i' = y_i/k^{1/2}$。

在这样的一次转换后，初始 2D 布局面积变为 S/K，平均单元密度为 K 的 2D 布局，后者将在合法化步骤中分布到 K 个器件层。Teris 型合法化（见 5.6.2.2 节）能应用以确定层指定，其也可能优化 TSV 数目和温度。如图 5-8 所示，一组互相堆叠的相邻单元在转换过程后分配到了不同的器件层。

5.5.2　折叠转换方法

LST 通过将相邻单元堆叠在一起实现了较短的线网长度。然而，大量的 TSV 将在本地线网单元堆叠时生成。如果目标 3D IC 工艺仅允许有限的 TSV 密度，就需要生成较少 TSV 的转换技术。

基于折叠的转换将初始 2D 布局像一张纸一样折叠而不砍掉布局的任何一部分。在任意两个单元间的距离将不增加且总线网长度确保减小。TSV 仅在跨越折叠线（在图 5-9 中以短线示出）的线网中引入。通过最小化线网长度的初始 2D 布局，这样的长线网数目应该相当小，这意味着折叠区域间的互连受到了限制，从而导致了少得多的 TSV（与 LST 转换结果比较，其中许多高密度本地互连穿过不同的器件层）。图 5-9a 示出了一种折叠的方法，称为 2 折叠，通过在 x 轴和 y 轴方向

各折叠一次实现。图 5-9b 示出了另一种折叠方法，称为 4 折叠，通过在 x 轴和 y 轴方向各折叠两次实现。折叠结果在 3D 布局中合法化，所以此步不需要合法化。

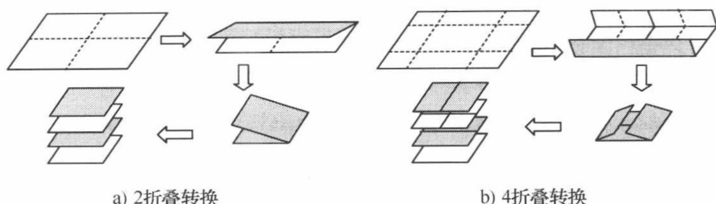


图 5-9　两种基于折叠的转换方法

在基于折叠的转换后，仅跨越折叠线的全局线网的长度（图 5-9 中的虚线）得到缩减。因此，基于折叠的转换技术不能达到 LST 的线网长度缩减程度。进一步讲，如果我们想保持芯片的初始宽高比，基于折叠的转换将受到奇数器件层的限制。

5.5.3　基于窗口的堆叠/折叠转换方法

如上所述，LST 方法以大量 TSV 的成本实现了线网长度最大缩减，而折叠方法使得 TSV 数较少，但线网长度较长并可能出现沿折叠线的高 TSV 密度。

理想的 3D 布局需要较短的线网长度，同时 TSV 密度需满足垂直互连工艺可支持的要求。此外，出于可布线度的考虑，我们倾向于较均匀的 TSV 密度。因此，我们提出了基于窗口的堆叠/折叠方法以实现更好的 TSV 密度控制。

在该方法中，2D 布局首先分为 $N \times N$ 个窗口。接着在每个窗口中使用堆叠或折叠转换。每个窗口可使用不同的堆叠/折叠顺序。图 5-10 示出了 $N=2$ 的情况。电路分为 2×2 个窗口（以实线示出）。每个窗口再次分为 4 个方块（以虚线示出）。每个方块中的数字代表该方块堆叠/折叠后所处器件层数。每个窗口以 4 层布局打包来构建最终的 3D 布局。

线网长度的缩减来源于以下原因：在同一块中的线网的长度保持不变；在同一窗口中的线网的长度极有可能受堆叠/折叠的影响而减小；同时穿过不同窗口线网长度减小。因此总线网长度质量得到提升。

与此同时，TSV 均匀分布在不同窗口并通过选择合理的层指定使解数目减少。TSV 由穿过相邻方块具有不同层数的线网引入，我们称该处于相邻方块间的边界为转边。较少的转边导致较少的 TSV 数目。因为我们需要将窗口内部方块分配到不同器件层，所以窗口内的转边不能减少。因此我们重点关注减少窗口间的转边。由于图 5-10a 中的顺序层指定产生了大量转边，我们使用如图 5-10b 所示的另一种层指定，称为对称指定，来将窗口间转边减小到零。故此层指定生成了最少的 TSV 数目，而线网长

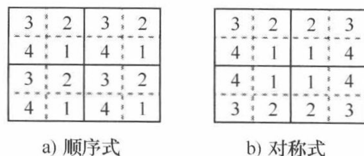


图 5-10　采用不同层指定的 2×2 窗口

度相似。

线网长度与 TSV 数目的折中可由窗口的数目来控制。

5.6 合法化和详细布局技术

在全局布局阶段不需要任意单元的最终位置。合法化负责去除单元间残存的重叠，同时详细布局执行布局质量的进一步的精细化。

粗合法化（见 5.6.1 节）桥接了全局布局和详细布局间的鸿沟。即使对于 5.2 节中讨论的离散基于布局技术，如果器件层数 K 不是 2 的幂，重叠也会在递归对分后出现。出于节约运行时间的目的，其他在 5.3 节和 5.4 节中讨论的连续技术通常在区域密度约束严格满足后停止。粗合法化将单元分布的更均匀，从而使随后的详细合法化阶段（见 5.6.2 节）能假设本地单元的移动足够得到合法布局。另外称为 Tetris 型合法化的合法化技术也将在 5.6.2.2 节中介绍。

详细布局执行了单元的本地交换以进一步精细化目标函数。如果在一个器件层内交换，其与 2D 详细布局没有区别。器件层间的交换是 3D 布局中的新内容。使用松弛冲突线网图来减少 TSV 数的技术将在 5.6.3 节介绍。

5.6.1 粗合法化

虽然在粗合法化后产生的布局仍含重叠，但单元均匀地分布在布局区域使详细布局中避免了在过大的区域内执行计算密集的本地计算。粗合法化[27]利用了称为单元平移的扩展的试探算法来为详细合法化和精细化布局做准备。

为了利用单元平移试探法，布局区域 $[0,W]\times[0,H]\times[0,K]$ 分为 $M\times N\times L$ 个桶，其中在（x_i，y_i，z_i）的单元 cell_i 映射到区域$[x_i-w_i/2,x_i+w_i/2]\times[y_i-h_i/2,y_i+h_i/2]\times[z_i-1,z_i]$。在单元平移过程中，单元每沿一个方向移动，即会向 3 个方向移动 3 次。

单元平移在 x 轴方向的展示在图 5-11 中示出。在该例中，行方向灰色的桶边界根据桶密度移动。标在桶中的数字是桶的密度，其中 d_{old} 和 d_{new} 分别是单元平移前后的桶密度。新的桶宽度 W'_b 和旧的桶宽度 W_b 之间的比率为 W'_b/W_b，对应此行从左至右的桶的比率分别约为 0.9、1.4、1.0、0.8、1.3 和 0.5。因此在这些桶中的单元也沿 x 轴方向移动并且桶的密度调整到满足密度约束。比率 W'_b/W_b 与桶密度 d 相关，如图 5-12 所示。在该图中，x 轴代表桶密度 d，y 轴表示比率 W'_b/W_b。系数 a_U、a_L 和 b 对每行内部均相同（与灰色行相同），但是对不同行可能有差异，其值被调整到能够保持每行中总的桶宽度为常数。

在单元平移后，每个桶中的单元密度确保不超过其容量。但该试探法无法考虑需优化的目标函数。因此，单元移动和单元交换操作在单元平移之后进行；这将优化目标函数式（5-8）并保持密度低于每个桶中的特性要求。

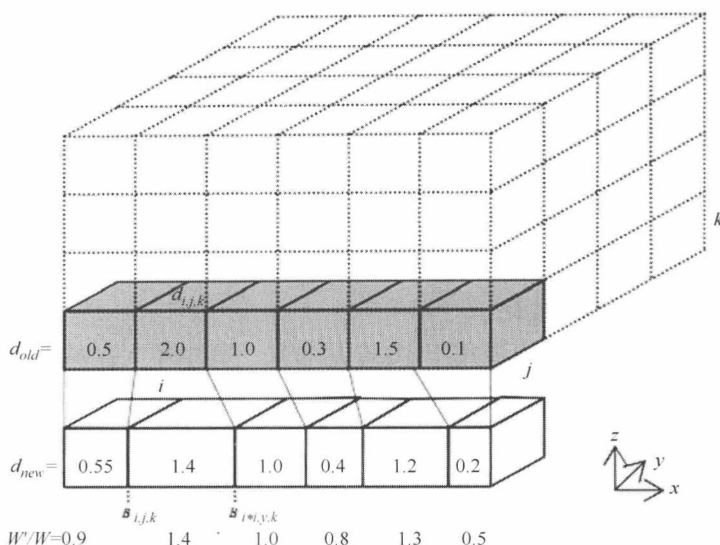


图 5-11　单元在 x 方向的平移[27]

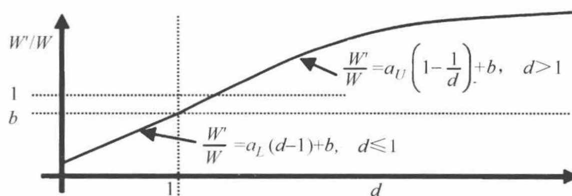


图 5-12　单元平移桶宽度与密度的对比[27]

5.6.2　详细合法化

详细合法化将单元置于最近的可用空间，实现目标函数最小退化。我们介绍两种实现该任务的详细合法化技术。基于 DAG 的合法化假设单元分布已经在粗合法化中实现了均匀化，并努力仅在附近移动单元。Tetris 型合法化则仅假设单元分布在（x，y）平面的投影中均匀，并能在层指定未给定时将其确定，或在初始层指定给定时最小化错位。

5.6.2.1　基于 DAG 的合法化

该详细合法化过程创建了一个比在粗合法化中使用的网格更精细的网格，并由相比平均单元面积更小的桶组成。通过用桶宽度除以桶中单元宽度（而不是面积）的准确值，桶密度实现了以更细化的计算。为了保证密度精确的在布局的不同半间平衡，由桶边缘形成的分割面的每一侧现存空间量或缺少的空间量分别被计算得到。构建定向开环图（DAG），其中从含多余单元面积的桶向可以接受附加单元面

积的邻近桶创建定向边。通过 DAG，可以得到桶处理顺序的从属关系，同时单元被按顺序放置到其最终位置。此外，目标函数对于单元移动的灵敏度的估计也用来确定单元的处理顺序。使用该处理顺序，算法在单元初始位置周围的目标区域中为每一个单元寻找最优可用位置。目标函数用于确定哪个目标区域中的可用位置可得到最优结果。如果可用位置没有找到，目标区域就逐渐扩展直到其中含的行碎片中有足够多的空闲空间。如果已处理的单元需要移开以合法放置单元，它们的移动在目标函数上的影响包含在将单元放置到该位置的成本中。

5.6.2.2 Tetris 型合法化

Tetris 型合法化技术[19]可应用于3D 全局布局，其中单元区域在（x，y）平面上的投影可得到较好分布。为准备合法化，所有的单元按其 x 轴坐标增序排序。从最左侧的单元开始，用类似2D 布局合法化[30]中使用方法的方式逐个确定单元的位置。每一次，考虑每层每行最左侧的合法位置。我们通过最小化重定位成本 R 选取位置：

$$R=\alpha\cdot d+\beta\cdot v+\gamma\cdot t \tag{5-41}$$

式中，d 是从全局布局结果得到的单元错位；v 是 TSV 数目；t 是热成本；系数 α、β、γ 为预设定权重；成本 d 与单元的（x，y）位置有关，同时成本 v 和 t 与单元的层指定有关。

在该合法化过程中，温度优化通过单元的层指定实现。在现有 3D IC 工艺[40]下，热沉通常贴在 3D IC 堆叠的底侧（和/或顶侧），而其他边缘是隔热的。因此 3D IC 堆叠中的主要热流是垂直于热沉的。本章参考文献［17］中的研究显示单元的 z 位置将比（x，y）位置对最终温度有更大的影响。因此，如果初始 2D 布局是热敏感的，就可以处理横向热流，从而使热单元均匀分布以避免热点。

全阻热模型用于最终温度验证。在优化过程的内部循环中，采用了用于温度优化的更简单和迅速的热模型[17]来加速布局过程。每个片堆叠视为独立的热阻链。每个片堆叠的最高温度即可写为

$$T=\sum_{i=1}^{k}\left(R_i\sum_{j=i}^{k}P_j\right)+R_b\sum_{i=1}^{k}P_i=\sum_{i=1}^{k}P_i\left(\sum_{j=1}^{i}R_j+R_b\right) \tag{5-42}$$

除速度更快外，这样的简单封闭等式还可为热敏感单元层指定提供直接向导。式（5-42）告诉我们片堆叠的最高温度是每层功率数的加权和。而每层的权重是该层以下阻值之和。靠近热沉的器件层权重较小。

指定单元 j 到层 i 的热成本 $t_{i,j}$ 写为

$$t_{i,j}=P_j\left(\sum_{k=1}^{i}R_k+R_b\right) \tag{5-43}$$

层指定的热成本在式（5-41）和布局精细化过程中均被使用，将在 5.6.3 节中介绍。

5.6.3　通过 RCN 图的层指定

在 5.5 节中提出的 3D 转换中，单元的层指定是基于简单试探法的。为了进一步减少 TSV 数目和温度，本章参考文献［19］中提出了一种创新的层指定算法来重新指定单元层。

5.6.3.1　冲突线网图 CNG

本章参考文献［8］中提出的金属线层指定算法扩展到了 3D 布局中的单元层指定。对于给定的合法化的 3D 布局，创建了冲突线网图，如图 5-13 所示，其中单元和通孔都是节点。每个线网指定一个通孔节点。有两种类型边：线网边和冲突边。星形模式下，每个线网中所有的单元通过线网边连接到通孔节点。如果相互重叠的单元置于同一层，那么在两者间创建冲突边。

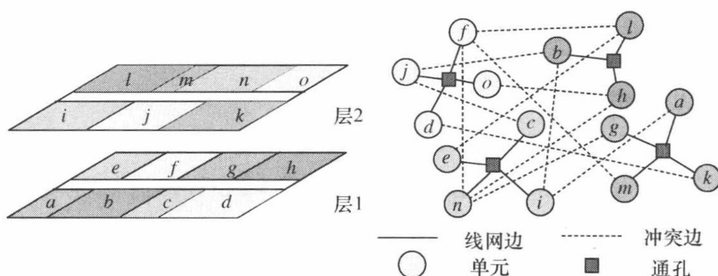


图 5-13　松弛后的冲突线网图

图 5-13 中对每个单元的层指定是首选方案，因为这样能将包含边成本和节点成本在内的总成本最小化。所有的线网边指定为成本 0。如果通过一个冲突边连接两个单元指定到同一层，冲突边的成本设置为 $+\infty$；相反，成本设为 0。通孔节点的成本是该通孔的高度，其代表了该线网中的总 TSV 数目。通孔的高度由连接它们的单元的层决定。单元节点 v_j 的成本是将 v_j 指定到 i 层的热成本 $t_{i,j}$。路径成本是沿该路径的边成本和节点成本的总和。

由此得到的图是一个有向开环图。动态编程优化法可用于在线性时间内找到图的每个引出的子树的最优解。接着，构建 CNG 中最大引出子树序列的算法用来覆盖大部分的初始图。最终引出子树的平均节点数最多能达到图中总节点数的 40%～50%。在子树的迭代优化后，我们可以得到全局优化解。具体的用 CNG 解层指定问题的详细算法请参阅本章参考文献［8］。

5.6.3.2　松弛非重叠约束

为了进一步减少 TSV 数目和最大片上温度，非重叠约束可松弛化来允许少量的重叠 r 以换取单元的层指定的更多自由度。

松弛化的非重叠定义如下：

$$重叠(i,j)=\begin{cases}假, & 如果\dfrac{o(i,j)}{s(i)+s(j)}\leqslant r\\ 真, & 如果\dfrac{o(i,j)}{s(i)+s(j)}>r\end{cases} \tag{5-44}$$

式中，$o(i,j)$ 为单元 v_i 和 v_j 重叠区域的面积；$s(i)$ 是单元 i 的面积。松弛基准 r 是 0 ~0.5 之间的正实数，均在图 5-14 中示出。因此，通过松弛化的非重叠约束，层指定结果将不再是合法化的 3D 布局。将需要另一轮合法化来消除重叠。

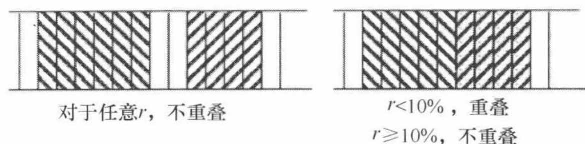


图 5-14 非重叠约束的松弛化

5.7 三维布局流程

3D 布局流程分为全局布局阶段、粗合法化阶段和详细合法化阶段，其中我们重点关注处在最前阶段的全局布局技术。

我们可使用 5.2 ~5.6 节中讨论的用于全局布局的基于分块的技术、二次均匀性建模技术、解析技术（作为多层引擎技术示例引入）或基于转换的技术。为了加速运行时间并得到更好的质量，可应用多层技术，其中任意一种以上的全局布局技术均可用作布局引擎。

粗合法化并非总是必要的，其应用依赖于详细合法化的要求。基于 DAG 的详细合法化需要在给定桶的粗略均匀密度分布，因此粗合法化在全局布局结果不能达到面积分配要求时是必要的。Tetris 型合法化工作适用于任意给定布局，但仍更倾向于为得到更优合法化的布局质量而进行均匀优化得到的全局布局。

在详细合法化后，像逐层 2D 详细布局一样，将会执行基于 RCN 的层指定精细化。在基于 RCN 精细化过程中如果重叠（例如 10%）被允许，合法化将被执行。多次 RCN 精细化和合法化的迭代后，布局质量持续改善。在合法的 3D 布局实现后，整个 3D 布局流程结束。

5.8 多种三维布局技术的影响

在本节中我们将总结多种 3D 布局技术的实验结果。

5.8.1 节包括在线网长度和 TSV 优化上得到的结果。基于转换的技术和多层解析布局技术在线网长度和 TSV 数目间折中的能力得以展示并相互比较。基于分块技术的结果也从本章参考文献［27］中提取并转换后做比较。读者还可以参阅本

章参考文献［41］了解二次均匀性建模布局技术的结果。在详细布局中，RCN 基于图的精细化也对线网长度和 TSV 数的折中有影响，其结果也会被说明。

5.8.2 节关注 3D 布局过程中的热优化。热线网权重和热敏感 Tetris 型合法化的实验结果在该节示出。

5.8.1　线网长度和 TSV 数目的折中

表 5-1 列出了基准电路[43]中的 18 种电路，用来测试 3D 布局器[26,27,42,19,13]。我们将使用该基准电路来在非热敏感的情况下比较 3D 布局结果。通过计算几何均值来测量和比较全部结果。

表 5-1　基准电路特征和 mPL[7]的 2D 布局结果

电　　路	#cell	#net	2D WL($\times 10^7$)
ibm01	12282	11507	0.47
ibm02	19321	18429	1.35
ibm03	22207	21621	1.23
ibm04	26633	26163	1.50
ibm05	29347	28446	3.50
ibm06	32185	33354	1.84
ibm07	45135	44394	2.87
ibm08	50977	47944	3.14
ibm09	51746	50393	2.61
ibm10	67692	64227	5.50
ibm11	68525	67016	3.88
ibm12	69663	67739	6.62
ibm13	81508	83806	4.92
ibm14	146009	143202	11.15
ibm15	158244	161196	12.51
ibm16	182137	181188	16.22
ibm17	183102	180684	25.76
ibm18	210323	200565	18.35
Geo-mean	—	—	4.15

我们首先在非热敏感的情况下比较多种基于变换的布局技术（见 5.5 节）的结果，见表 5-2。结果从不同的变换方案中生成，包括本地堆叠转换“LST”，基于窗口的转换“LST（8×8 窗）”，折叠转换“2 折叠”。LST 和 2 折叠与 5.5.1 节和 5.5.2 节中描述的相同，LST（8×8 窗）是通过将布局区域分割为 8×8 个窗口并

在每个窗口中进行 LST 的基于窗口的转换。与 2 折叠相比，LST 能以 17 倍 TSV 数目的成本减小 44% 的线网长度；LST（8×8 窗）能以 5 倍 TSV 数目的成本减小 20% 的线网长度。这些结果展示了基于转换方法在线网长度和 TSV 数间折中的能力，其可通过在基于窗口的混合变换中改变窗口数目实现。转换方法的选择依赖整体线网长度的重要性和 TSV 制造成本。

表 5-2　基于转换技术的三维布局结果

电　路	LST		LST(8×8win)		2 折叠	
	WL($\times 10^7$)	#TSV($\times 10^3$)	WL($\times 10^7$)	#TSV($\times 10^3$)	WL($\times 10^7$)	#TSV($\times 10^3$)
ibm01	0.24	21.03	0.34	6.69	0.43	1.57
ibm02	0.66	33.31	0.85	14.60	1.25	3.09
ibm03	0.61	36.38	0.79	12.73	1.03	3.38
ibm04	0.76	44.95	1.01	15.63	1.35	3.63
ibm05	2.36	50.67	2.63	25.85	3.99	8.17
ibm06	0.94	57.91	1.28	18.61	1.69	3.26
ibm07	1.46	77.60	2.03	25.16	2.65	5.81
ibm08	1.59	83.50	2.21	26.00	2.91	5.03
ibm09	1.34	87.44	2.03	22.92	2.41	4.33
ibm10	2.80	116.92	4.22	32.52	5.08	5.67
ibm11	2.00	117.03	3.05	29.25	3.55	6.01
ibm12	3.36	124.61	4.78	39.67	6.19	6.49
ibm13	2.53	144.73	3.83	34.26	4.55	6.61
ibm14	5.70	247.46	8.93	56.67	10.34	9.45
ibm15	6.40	284.74	9.91	59.55	11.29	12.07
ibm16	8.30	326.99	13.38	73.66	15.04	12.53
ibm17	13.16	332.80	19.51	92.66	24.21	14.53
ibm18	9.37	359.07	14.81	75.27	16.43	14.19
Geo-mean	2.14	101.44	3.07	29.65	3.84	5.95

表 5-3 展示了 TSV 权重 $\alpha_{TSV}=10$ 的多层解析技术（见 5.4 节）。共收集了几套结果：对于单层布局、两层布局、三层布局的结果及使用逐层 2D 布局的详细布局结果。单层布局用于直接执行解析布局引擎而不进行任何分簇，而两层布局和三层布局通过分簇构建了两层和三层架构。在这些结果中，我们看到在相同的 TSV 数目权重下，单层布局实现了最短线网长度，而三层布局实现了最少的 TSV 数目。我们通过比较采用 LST($r=10\%$)（最优线网长度情况）的单层布局、采用 LST（8×8 窗）的两层布局、采用 2 折叠（最优 TSV 情况）的三层布局来比较多层解析布

局技术和基于变换的布局技术。从表 5-2 和表 5-3 中的数据来看，很明显单层布局相对 LST($r=10\%$) 平均能实现 29% 的 TSV 数目缩减而仅有 5% 的线网长度恶化；三层布局相对 2 折叠也能实现 12% 的线网长度缩减和 24% 的 TSV 数目缩减。

表 5-3　基于多层布局技术的三维布局结果

电　路	1 层		2 层		3 层	
	WL($\times10^7$)	#TSV($\times10^3$)	WL($\times10^7$)	#TSV($\times10^3$)	WL($\times10^7$)	#TSV($\times10^3$)
ibm01	0.28	8.12	0.37	1.28	0.37	1.09
ibm02	0.73	15.82	1.13	2.26	1.04	3.08
ibm03	0.67	16.67	0.79	3.51	0.89	2.21
ibm04	0.82	28.79	1.12	5.04	1.22	2.17
ibm05	1.88	31.77	2.15	13.20	2.50	9.04
ibm06	1.01	38.17	1.31	6.89	1.54	2.86
ibm07	1.56	54.21	2.05	9.03	2.40	3.33
ibm08	1.69	53.71	2.07	11.64	2.39	4.32
ibm09	1.44	61.65	1.84	10.73	2.24	2.73
ibm10	2.90	88.62	3.90	18.16	4.60	3.79
ibm11	2.12	88.46	2.70	16.16	3.19	4.07
ibm12	3.59	95.89	4.82	19.51	5.80	4.59
ibm13	2.68	110.56	3.35	21.00	4.05	4.12
ibm14	5.95	219.65	6.76	73.71	9.39	9.95
ibm15	6.67	260.17	7.56	84.33	10.36	9.74
ibm16	8.42	300.69	9.58	106.46	13.89	9.89
ibm17	13.28	310.52	15.49	120.77	20.59	12.29
ibm18	9.52	333.75	10.89	107.49	14.60	12.58
Geo-mean	2.24	70.78	2.80	16.12	3.36	4.55

表 5-4 展示了基于分块技术（见 5.2 节）在不同 TSV 权重下的结果。这些数据从本章参考文献［25］的结果转化而来，其为基于基准电路[43]的变形版本。在本章参考文献［25］中，行间距设为 25% 的行高，而在原始基准电路中行间距等于行高。为得到与表 5-2 和表 5-3 可比较的数据，我们假设本章参考文献［25］中的线网长度具有等量的 X 方向线网和 Y 方向线网，使用系数 $50\%+50\%\times2/(1+25\%)=1.3$ 来按比例缩小线网长度。表 5-4 中的 3 列具有不断增加的 TSV 权重，其中也示出了线网长度和 TSV 数目间的折中。最右侧的列 TSV 数目最佳，比表 5-3 中的三层布局节省了 40% 的 TSV 数目但产生了 33% 的线网长度恶化。但最左侧具有最优线网长度的列比表 5-3 中的单层布局多消耗了 20% 的线网长度和 39% 的

TSV 数目。中间列也不能像表 5-3 中的两层工作得一样好。这些数据说明由于基于分块的技术的分块特性，其擅长 TSV 数目的缩减，但它们不像多层技术那样适用于制造更多 TSV 以实现更短线网长度的情况。

表 5-4　基于分块布局技术的三维布局结果

电　路	8.00E-07		2.00E-04		1.30E-02	
	WL($\times 10^7$)	#TSV($\times 10^3$)	WL($\times 10^7$)	#TSV($\times 10^3$)	WL($\times 10^7$)	#TSV($\times 10^3$)
ibm01	0.30	20.50	0.39	5.39	0.52	0.49
ibm02	0.85	32.44	0.97	11.98	1.51	0.86
ibm03	0.81	34.87	0.95	9.97	1.23	1.95
ibm04	1.02	42.43	1.13	14.24	1.61	2.05
ibm05	2.18	49.78	2.29	20.29	3.07	5.92
ibm06	1.34	55.35	1.47	20.29	2.09	2.47
ibm07	1.91	74.51	2.15	24.77	3.23	2.85
ibm08	2.06	80.86	2.32	26.39	3.35	2.59
ibm09	1.78	83.96	2.10	24.97	2.94	1.79
ibm10	3.33	115.48	3.82	35.25	5.80	2.39
ibm11	2.60	112.90	3.01	33.59	4.31	2.69
ibm12	4.44	121.39	4.89	44.50	7.52	3.97
ibm13	3.26	139.26	3.78	41.85	5.63	2.63
ibm14	7.16	238.96	7.82	80.71	12.23	4.16
ibm15	8.29	275.91	9.10	91.86	13.20	6.40
ibm16	10.43	319.76	11.52	105.99	18.44	5.58
ibm17	15.20	327.27	16.37	125.42	26.13	7.59
ibm18	11.21	350.36	12.41	110.94	19.98	4.58
Geo-mean	2.70	98.27	3.04	32.32	4.48	2.80

如 5.6.3 节所述，基于 RCN 图的层指定过程[19]用来进一步优化 3D 电路的 TSV 数目。表 5-5 和表 5-6 分别示出了基于 RCN 图的层指定算法用在通过本地堆叠转换（见 5.5.1 节）而实现的布局和通过平面解析技术（见 5.4.2 节）实现的布局的效果。允许重叠为 $r=0$ 和 $r=10\%$ 的 RCN 精细化结果给出，其中 $r=0\%$ 是严格非重叠约束而 $r=10\%$ 允许精细化过程中相邻单元 10% 的重叠。在表 5-5 中，$r=0\%$ 时平均 TSV 缩减为 4%，线网长度无任何恶化；$r=10\%$ 时平均 TSV 缩减为 15%，线网长度恶化极小。在表 5-6 中，$r=0$ 时平均 TSV 缩减为 3%，线网长度无任何恶化；$r=10\%$ 时平均 TSV 缩减为 9%，线网长度恶化 1%。从这些结果可以看出，本地堆叠转换的布局比打平解析布局有更多的改善空间，这也说明解析布局方

法得到了比基于转换的布局方法更好的解。

表 5-5　本地堆叠结果和在 $r=0$ 和 10% 时的 RCN 优化结果

电　路	LST		$r=0$ 的 RCN 后		$r=10\%$ 的 RCN 后	
	WL($\times 10^7$)	#TSV($\times 10^3$)	WL($\times 10^7$)	#TSV($\times 10^3$)	WL($\times 10^7$)	#TSV($\times 10^3$)
ibm01	0. 24	21. 03	0. 24	20. 73	0. 24	18. 63
ibm02	0. 66	33. 31	0. 66	32. 75	0. 66	28. 87
ibm03	0. 61	36. 38	0. 62	35. 38	0. 62	30. 49
ibm04	0. 76	44. 95	0. 76	43. 44	0. 77	38. 07
ibm05	2. 36	50. 67	2. 36	48. 82	2. 36	44. 37
ibm06	0. 94	57. 91	0. 94	57. 29	0. 95	50. 26
ibm07	1. 46	77. 60	1. 46	74. 35	1. 47	64. 85
ibm08	1. 59	83. 50	1. 59	78. 42	1. 59	70. 46
ibm09	1. 34	87. 44	1. 33	82. 79	1. 35	73. 13
ibm10	2. 80	116. 92	2. 80	112. 62	2. 81	99. 59
ibm11	2. 00	117. 03	2. 00	112. 29	2. 02	98. 77
ibm12	3. 36	124. 61	3. 37	121. 31	3. 38	107. 89
ibm13	2. 53	144. 73	2. 53	138. 41	2. 54	122. 95
ibm14	5. 70	247. 46	5. 70	234. 24	5. 73	210. 08
ibm15	6. 40	284. 74	6. 40	267. 28	6. 41	248. 06
ibm16	8. 30	326. 99	8. 30	311. 33	8. 34	283. 10
ibm17	13. 16	332. 80	13. 16	320. 34	13. 15	286. 26
ibm18	9. 37	359. 07	9. 39	337. 12	9. 40	300. 87
Geo-mean	2. 14	101. 44	2. 14	97. 46	2. 15	86. 73

表 5-6　打平解析结果和在 $r=0$ 和 10% 时的 RCN 优化结果

电　路	1 层		$r=0$ 的 RCN 后		$r=10\%$ 的 RCN 后	
	WL($\times 10^7$)	#TSV($\times 10^3$)	WL($\times 10^7$)	#TSV($\times 10^3$)	WL($\times 10^7$)	#TSV($\times 10^3$)
ibm01	0. 28	8. 12	0. 28	8. 03	0. 29	7. 87
ibm02	0. 73	15. 82	0. 73	15. 69	0. 76	15. 59
ibm03	0. 67	16. 67	0. 67	16. 45	0. 69	16. 10
ibm04	0. 82	28. 79	0. 82	27. 99	0. 84	26. 56
ibm05	1. 88	31. 77	1. 88	30. 94	1. 89	30. 20
ibm06	1. 01	38. 17	1. 01	37. 24	1. 04	35. 58
ibm07	1. 56	54. 21	1. 56	52. 82	1. 59	49. 57
ibm08	1. 69	53. 71	1. 69	52. 66	1. 71	50. 97

（续）

电　　路	1 层		$r=0$ 的 RCN 后		$r=10\%$ 的 RCN 后	
	WL（×10^7）	#TSV（×10^3）	WL（×10^7）	#TSV（×10^3）	WL（×10^7）	#TSV（×10^3）
ibm09	1.44	61.65	1.44	59.88	1.47	56.37
ibm10	2.90	88.62	2.90	86.26	2.97	81.19
ibm11	2.12	88.46	2.12	85.39	2.15	79.63
ibm12	3.59	95.89	3.59	93.51	3.64	87.73
ibm13	2.68	110.56	2.68	106.74	2.71	99.67
ibm14	5.95	219.65	5.95	209.11	5.92	188.71
ibm15	6.67	260.17	6.67	246.45	6.62	224.01
ibm16	8.42	300.69	8.42	288.13	8.35	261.84
ibm17	13.28	310.52	13.28	297.61	13.18	267.90
ibm18	9.52	333.75	9.52	318.80	9.45	286.02
Geo-mean	2.24	70.78	2.24	68.67	2.27	64.61

5.8.2 热优化的影响

5.8.2.1 热敏感线网权重对于温度的影响

式（5-6）中定义的热敏感项用于在线网长度优化过程中控制温度。一个大的热系数 α_{TEMP} 以更长线网长度和更多 TSV 数目为成本降低了更多的温度。式（5-12）中定义的热敏感线网权重是实现热敏感的等效方法，其与热系数 α_{TEMP} 成正比。

热敏感线网权重在基于分块的 3D 布局器[27]中实现，其对温度降低的影响和对线网长度及 TSV 数目的影响如图 5-15 所示。对于稍许改动的基准电路[43]也进行了实验。TSV 系数设为 10（μm），计算了热系数对 TSV 数（层间通孔数）、线网长度、总功耗、平均温度和最大温度的影响。当与未加权结果比较时，这些方面的改变百分比取基准电路中 ibm01 ~ ibm18 的平均改变百分比。当平均温度减小 19%

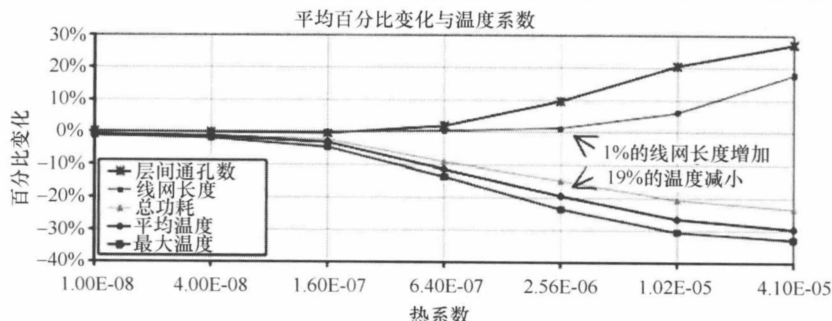


图 5-15　在热系数变化时的平均变化百分比[27]

时，线网长度仅增加 1%，TSV 数目增加 10%。

5.8.2.2　合法化对温度的影响

这里我们比较两个 Tetris 型合法化过程：一个不含热敏感处理，一个包含热敏感处理。单元功耗通过在 $10^5 \sim 10^6 W/m^2$ 的范围内指定单元功耗密度随机确定[39]。温度求值采用热阻网络模型和本章参考文献［40］中的热阻值。初始布局通过应用基于转换的技术（见 5.5 节）中的本地堆叠（LST）法生成。结果在表 5-7 中示出，表中温度为最大片上温度和热沉温度的差值。与不含热敏感处理的合法化相比，热敏感合法化平均能降低 39% 的最大片上温度，带来 8% 的线网长度增加，节省 5% 的 TSV 数目。

表 5-7　Tertis 式合法化的热敏感结果

电　路	非热敏感的 Tetris 式合法化			热敏感的 Tetris 式合法化		
	WL($\times 10^7$)	#TSV($\times 10^3$)	温度/℃	WL($\times 10^7$)	#TSV($\times 10^3$)	温度/℃
ibm01	0.24	21.03	279.002	0.29	19.67	150.422
ibm02	0.66	33.31	207.802	0.72	31.83	117.516
ibm03	0.61	36.38	205.766	0.67	34.13	120.487
ibm04	0.76	44.95	163.279	0.85	42.05	94.648
ibm05	2.36	50.67	138.501	2.44	48.59	78.607
ibm06	0.94	57.91	165.881	1.05	52.12	101.269
ibm07	1.46	77.60	108.015	1.57	72.93	68.382
ibm08	1.59	83.50	101.04	1.68	78.86	61.897
ibm09	1.34	87.44	96.899	1.47	83.35	59.7815
ibm10	2.80	116.92	58.335	3.01	112.95	36.3501
ibm11	2.00	117.03	283.705	2.18	108.96	172.396
ibm12	3.36	124.61	206.811	3.65	120.89	122.211
ibm13	2.53	144.73	254.684	2.76	134.61	157.983
ibm14	5.70	247.46	128.623	6.07	235.17	83.4365
ibm15	6.40	284.74	137.455	6.76	274.44	87.672
ibm16	8.30	326.99	98.5005	8.74	318.43	62.428
ibm17	13.16	332.80	84.73	13.62	324.44	52.954
ibm18	9.37	359.07	89.203	9.76	348.26	57.089
Geo-mean	2.14	101.44	141.88	2.32	96.30	86.11

5.9　三维布局对线网长度和中继器使用的影响

本节中我们将介绍多个针对 3D IC 技术对线网长度和中继器使用的影响的研究

成果[20]。线网长度采用半周长模型，中继器使用通过互连优化器 IPEM[14] 在后布局/预布线阶段估算得到，其中 2D 和 3D 布局通过先进的 2D 布局器 mPL6[7] 和多层解析 3D 布局器[13] 得到。在基准电路套装[43] 上的实验显示，中继器的总数在 3 层和 4 层 3D 电路上分别比 2D 电路减少 22% 和 50%。

5.9.1 二维/三维布局器和中继器估计

mPL6[7] 是大规模混合尺寸布局，包含多层解析布局器和鲁棒合法化器及详细布局器。这是为线网长度驱动的布局设计的，同时对密度敏感。ISPD 2006 布局竞赛[34] 显示 mPL6 在所有参加比赛的布局器中实现了最优的线网长度。

为了利用 3D 技术的优点，我们使用多层解析 3D 布局器（见 5.4 节）。该 3D 布局器考虑了线网长度和 TSV 数的折中，并显示了比基于转换的技术和基于分块的技术更好的折中能力。请参照 5.8.1 节中更多的实验结果。

IPEM[14] 提供了一系列程序来估计在深亚微米工艺下多种性能优化后的互连性能。这些优化算法包括 OWS（最优线网尺寸调整）、SDWS（同步驱动器和线网尺寸调整）、BIWS（缓冲器插入和线网尺寸调整）和 BISWS（缓冲插入、尺寸调整和线网尺寸调整）。同时大量的互连版图优化工具（如 Trio[11]，IPEM）均瞄准快速准确地估计优化后的互连延迟和面积，从而可通过使用简单封闭计算过程尽可能早地实现设计收敛。试验结果[14] 显示 IPEM 以比 Trio 快 1000 倍的速度运行，可达到的平均准确度为 90%。

5.9.2 实验设置和结果

实验在 IBM 布局基准电路[43] 上进行。由于这些基准电路没有输出/输入引脚信息，为得到相对更准确线网长度信息，我们使用一个线网的最小线网长度树来估计该线网需要的优化中的继器数目。

直线 Steiner 最小树广泛地应用于设计的早期阶段，如物理综合、布局规划、互连规划和布局中以估计线网长度、布线拥塞、互连延迟。它使用最小线网长度边来连接给定线网的节点。使用直线 Steiner 树构建包 FLUTE[10] 计算 Steiner 线网长度树来估计不需执行详细布线的中继器插入。FLUTE 基于预计算的查找表来使低阶线网的 Steiner 最小树快速准确构建。对于高阶线网，其被分为多个低阶线网直到可使用查找表。

为了准确估计 TSV 电阻和电容的延迟和面积，本章参考文献［22］中的方法用来将 TSV 建模为一定长度的线网。因为其巨大的尺寸，TSV 自电容较大。通过每个通孔和每层中第二层金属线长度的仿真，作者[22] 估算 3μm 厚的 TSV 相当于 8 ~ 20μm 长的金属线。因为每个 TSV 较大的纵切面积（每个 TSV 约 0.1Ω），其电阻不那么大，约等价于 0.2μm 长的第二层金属线。我们使用 MIT 林肯实验室开发的 3D 工艺，其相邻层间的最小距离是 2 ~ 3.45μm。因此，我们能大概将所有的相

邻层间的 TSV 等效为 14μm 长的金属线（8～20μm 间的平均值）。该值将在 TSV 穿过两层时翻番。

由于 FLUTE 仅能生成 2D 最小线网长度树，为了将其转化用于 3D 设计的 3D 树，需要做以下假设：①假设所有的树线网置于 3D 堆叠层的中间层；②在其他层中的引脚使用 TSV 连接到中间层的树。该假设最小化了线网中的总传统线网但高估了 TSV 总数。因此，相比本章参考文献［19］中的使用的仅简单将通孔数设为层数和线网范围设定 TSV 和线网长度估计方法，我们可以得到关于总线网长度更准确的信息。

实验在 32nm 工艺下进行。用来配置 IPEM 的工艺参数在表 5-8 中列出，并且我们运行了 FLUTE 和 IPEM 对每个基准电路中的每条线网。

表 5-8　工艺参数

工艺	32nm
时钟频率	2GHz
供电电压（V_{DD}）	0. 9V
最小尺寸中继单元大小（W_{min}）	70nm
晶体管输出电阻（r_g）	5kΩ
晶体管输出电容（c_p）	0. 0165fF
晶体管输入电容（c_g）	0. 105fF
金属线单位长度电阻（r）	1. 2Ω/μm
金属线单位面积电容（c_a）	0. 148fF/μm^2
金属线有效边缘电容（c_f）	0. 08fF/μm

表 5-9 给出了 2D 设计、三层 3D 设计、4 层 3D 设计和 IBM 布局基准电路的比较。该表列出了每个电路的线网长度（*WL* 单位 μm）和中继器数（#repeater），还有整体几何均值和归一化几何均值。由结果可见，通过应用三层 3D 设计，平均下来，相比 2D 设计的情况，总线网长度可减少 17%，互连中的中继器数目可减少 22%。此外，当使用四层的 3D 设计时，线网长度可进一步减小 40%，中继器数目可减少 50%。

表 5-9　IBM 布局基准电路的线网长度/中继器结果

电　路	2D 设计		3 器件层的 3D 设计		4 器件层的 3D 设计	
	WL(×10^7)	#中继器 (×10^3)	WL(×10^7)	#中继器 (×10^3)	WL(×10^7)	#中继器 (×10^3)
ibm01	0. 54	5. 26	0. 52	4. 80	0. 37	2. 85
ibm02	1. 58	18. 36	1. 62	18. 81	0. 96	9. 49
ibm03	1. 40	15. 65	1. 11	11. 52	0. 85	7. 75

（续）

电　路	2D设计		3器件层的3D设计		4器件层的3D设计	
	WL（$\times 10^7$）	#中继器（$\times 10^3$）	WL（$\times 10^7$）	#中继器（$\times 10^3$）	WL（$\times 10^7$）	#中继器（$\times 10^3$）
ibm04	1.65	17.69	1.40	14.04	1.02	8.83
ibm05	4.08	51.81	3.09	37.80	2.35	27.21
ibm06	2.16	23.72	1.89	19.72	1.33	12.13
ibm07	3.18	35.61	2.72	29.01	1.94	17.88
ibm08	3.71	42.95	3.22	35.54	2.23	21.80
ibm09	2.94	31.54	2.58	26.07	1.84	15.85
ibm10	6.09	72.10	5.27	60.09	3.52	35.48
ibm11	4.22	45.33	3.83	39.36	2.58	22.09
ibm12	7.42	89.33	6.29	73.05	4.37	46.05
ibm13	5.50	60.63	4.26	42.51	3.34	29.97
ibm14	12.22	141.59	9.36	101.05	7.04	68.48
ibm15	13.88	162.04	10.27	110.37	8.03	80.01
ibm16	18.25	219.26	13.26	147.95	10.21	105.23
ibm17	28.26	358.37	21.31	258.89	15.32	173.60
ibm18	20.75	248.70	14.73	162.79	11.62	120.13
Geo-mean	4.67	53.43	3.87	41.76	2.79	26.74

通过3D IC与2D情况的比较，中继器数目的减少一直多于总线网长度的减小。这是因为增加的层数将有效减小带有较大最小线网长度树的线网的长度，而带有最小线网长度树的线网总是不需要中继器。如在IPEM结果中看到的那样，小于500μm的线通常生成零个中继器。因此，通过减少最小线网长度较大的树的线网长度，我们能有效减少中继器的数目和片上互连的面积/功耗。

5.10　总结和结论

三维IC工艺为电路设计了附加一个维度的自由空间。这将提升器件封装密度并减小总体互连的长度，因此将有利于3D电路的功能、性能和功耗的提升。然而，该技术也挑战了布局工具。TSV的制造并不简单，因此布局工具需要考虑TSV的成本并实现折中来避免抵消线网长度减小带来的益处。由于热源的堆叠和长的热耗散路径，热问题也是3D电路的关键挑战。

在本章中，我们给出了热敏感3D布局工具问题的定义和现有文献中3D布局技术的总结。我们特别介绍了几种具有代表性的3D布局技术的细节，包括基于分

块的技术、二次均匀建模技术、多层布局技术和基于转换的技术。特别针对 3D 布局的合法化和详细布局技术进行了介绍。

基于分块的技术在 5.2 节中介绍。这些基于分块的技术是在传统的基于分块过程中的一些合适阶段插入与器件层平行的分块面。分块的成本通过估计线网长度和 TSV 数的加权计算，其中线网进一步用热敏感或拥塞敏感系数加权以符合温度和可布线性的要求。

二次均匀建模技术属于二次布局技术的范畴，其由打平布局技术组成。由于未约束的二次布局将引入大量的单元重叠，研发了不同的变形形式来进行重叠去除。二次一致性建模技术[41]通过每次迭代的另一个二次函数增加了密度惩罚函数，这样整个全局布局将能通过最小化二次函数求解。

5.4 节中介绍的多层技术[13]构建了从初始网表得来的物理层次，并解决了从最粗糙层次到最精细层次的一系列布局问题。

在以上这些技术之外，5.5 节介绍的基于转换的技术利用了现存 2D 布局结果构建了通过转换得到的 3D 布局。

作为多种 3D 全局布局技术的补充，合法化和详细布局技术在 3D 布局过程中的特定联系在 5.6 节中讨论。

最终，给出的实验数据展示了多种 3D 布局技术对线网长度、TSV 数目和温度的有效性，同时展示了 3D IC 工艺对线网长度和中继器使用的影响。这些实验数据说明基于分块的 3D 布局技术擅长 TSV 数目最小化，但不像多层解析技术在可制造更多 TSV 的情况下对线网长度的优化那么有效。对于多层解析布局技术，通过更多布局优化的层次可以线网长度增加为成本实现更少的 TSV 数目。最终，RCN 基于图的层指定过程对 TSV 和热优化都有效。

致　谢

本研究分别由 Gigascale Silicon 研究中心、IBM DARPA 子协议、美国国家自然科学基金 CCF-0430077 和 CCF-0528583 支持。

参考文献

1. C. Ababei, H. Mogal, and K. Bazargan, Three-dimensional place and route for FPGAs, *Proceedings of the 2005 Conference on Asia South Pacific Design Automation*, pp. 773–778, 2005.
2. C. Alpert, A. Kahng, G.-J. Nam, S. Reda, and P. Villarrubia, A semi-persistent clustering technique for VLSI circuit placement, *Proceedings of the 2005 International Symposium on Physical Design*, pp. 200–207, 2005.
3. K. Balakrishnan, V. Nanda, S. Easwar, and S. K. Lim, Wire congestion and thermal aware 3D global placement, *Proceedings of the 2005 Conference on Asia South Pacific Design Automation*, pp. 1131–1134, 2005.

4. D. P. Bertsekas, Approximation procedures based on the method of multipliers, *Journal of Optimization Theory and Applications*, 23(4), 487–510, 1977.
5. T. F. Chan, J. Cong, T. Kong, and J. R. Shinnerl, Multilevel optimization for large-scale circuit placement, *Proceedings of the 2000 IEEE/ACM International Conference on Computer-aided Design*, pp. 171–176, 2000.
6. T. F. Chan, J. Cong, and K. Sze, Multilevel generalized force-directed method for circuit placement, *Proceedings of the 2005 International Symposium on Physical Design*, pp. 185–192, 2005.
7. T. F. Chan, J. Cong, J. R. Shinnerl, K. Sze, and M. Xie, mPL6: enhancement multilevel mixed-size placement with congestion control, in *Modern Circuit Placement*, G.-J. Nam and J. Cong, Eds., Springer, New York, NY, 2007.
8. C.-C. Chang and J. Cong, An efficient approach to multilayer layer assignment with an application to via minimization, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(5): 608–620, 1999.
9. T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, A high-quality mixed-size analytical placer considering preplaced blocks and density constraints, *Proceedings of the 2006 IEEE/ACM International Conference on Computer-Aided Design*, pp. 187–192, 2006.
10. C. Chu and Y. Wong, FLUTE: Fast lookup table based rectilinear steiner minimal tree algorithm for VLSI design, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(1): 70–83, 2008.
11. J. Cong and L. He, Theory and algorithm of local refinement based optimization with application to device and interconnect sizing, *IEEE Transactions on Computer-Aided Design*, pp. 1–14, 1999.
12. J. Cong and G. Luo, Highly efficient gradient computation for density-constrained analytical placement methods, *Proceedings of the 2008 International Symposium on Physical Design*, pp. 39–46, 2008.
13. J. Cong and G. Luo, A multilevel analytical placement for 3D ICs, *Proceedings of the 2009 Conference on Asia and South Pacific Design Automation*, Yokohama, Japan, pp. 361–366, 2009.
14. J. Cong and D. Z. Pan, Interconnect estimation and planning for deep submicron designs, *Proceedings of the 26th ACM/IEEE Design Automation Conference*, New Orleans, LA, pp. 507–510,\, 1999.
15. J. Cong and J. Shinnerl, *Multilevel Optimization in VLSICAD*, Kluwer Academic Publishers, Boston, MA, 2003.
16. J. Cong and M. Xie, A robust mixed-size legalization and detailed placement algorithm, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(8): 1349–1362, 2008.
17. J. Cong and Y. Zhang, Thermal via planning for 3-D ICs, *Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design*, pp. 745–752, 2005.
18. J. Cong, J. R. Shinnerl, M. Xie, T. Kong, and X. Yuan, Large-scale circuit placement, *ACM Transactions on Design Automation Electronic Systems*, 10(2): 389–430, 2005.
19. J. Cong, G. Luo, J. Wei, and Y. Zhang, Thermal-aware 3D IC placement via Transformation, *Proceedings of the 2007 Conference on Asia and South Pacific Design Automation*, pp. 780–785, 2007.
20. J. Cong, C. Liu, and G. Luo, Quantitative studies of impact of 3D IC design on repeater usage, *Proceedings of the International VLSI/ULSI Multilevel Interconnection Conference*, 2008.
21. S. Das, *Design Automation and Analysis of Three-Dimensional Integrated Circuits*, PhD Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2004.
22. W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, Demystifying 3D ICs: The pros and cons of going vertical, *IEEE Design & Test of Computers*, 22(6): 498–510,\, 2005.
23. A. E. Dunlop and B. W. Kernighan, A procedure for placement of standard-cell VLSI circuits, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 4(1): 92–98, 1985.

24. C. M. Fiduccia and R. M. Mattheyses, A linear-time heuristic for improving network partitions, *Proceedings of the 19th ACM/IEEE Conference on Design Automation*, pp. 175–181, 1982.
25. B. Goplen, *Advanced Placement Techniques for Future VLSI Circuits*, PhD Dissertation, University of Minnesota, Minneapolis, MN, 2006.
26. B. Goplen and S. Sapatnekar, Efficient thermal placement of standard cells in 3D ICs using a force directed approach, *Proceedings of the 2003 IEEE/ACM International Conference on Computer-Aided design*, p. 86, 2003.
27. B. Goplen and S. Sapatnekar, Placement of 3D ICs with thermal and interlayer via considerations, *Proceedings of the 44th Annual Conference on Design Automation*, pp. 626–631, 2007.
28. A. S. Grove, *Physics and Technology of Semiconductor Devices*, John Wiley & Sons, Inc., Hoboken, NJ, 1967.
29. R. Hentschke, G. Flach, F. Pinto, and R. Reis, 3D-vias aware quadratic placement for 3D VLSI circuits, *IEEE Computer Society Annual Symposium on VLSI*, pp. 67–72, 2007.
30. D. Hill, Method and system for high speed detailed placement of cells within an integrated circuit design, US Patent 6370673, 2001.
31. A. B. Kahng, S. Reda, and Q. Wang, Architecture and details of a high quality, large-scale analytical placer, *Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design*, pp. 891–898, 2005.
32. G. Karypis and V. Kumar, Multilevel k-way hypergraph partitioning, *Proceedings of the 36th ACM/IEEE Conference on Design Automation*, pp. 343–348, 1999.
33. I. Kaya, S. Salewski, M. Olbrich, and E. Barke, Wirelength reduction using 3-D physical design, *Proceedings of the 14th International Workshop on Power and Timing Optimization and Simulation*, pp. 453–462, 2004.
34. G.-J. Nam, ISPD 2006 placement contest: benchmark suite and results, *Proceedings of the 2006 International Symposium on Physical Design*, pp. 167–167, 2006.
35. G.-J. Nam and J. Cong (Eds.), *Modern Circuit Placement: Best Practices and Results*, Springer, New York, NY, 2007.
36. W. C. Naylor, R. Donelly, and L. Sha, Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer, US Patent 6301693, 2001.
37. J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, NY, 2006.
38. P. Spindler and F. M. Johannes, Fast and robust quadratic placement combined with an exact linear net model, *Proceedings of the 2006 IEEE/ACM International Conference on Computer-Aided Design*, pp. 179–186, 2006.
39. C.-H. Tsai and S.-M. Kang, Cell-level placement for improving substrate thermal distribution, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(2): 253–266, 2000.
40. P. Wilkerson, A. Raman, and M. Turowski, Fast, automated thermal simulation of three-dimensional integrated circuits, *Proceedings of the 9th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, Las Vegas, Nevada, 2004.
41. H. Yan, Q. Zhou, and X. Hong, Thermal aware placement in 3D ICs using quadratic uniformity modeling approach, *Integration, the VLSI Journal*, 42(2): 175–180,\, 2009.
42. B. Yao, H. Chen, C.-K. Cheng, N.-C. Chou, L.-T. Liu, and P. Suaris, Unified quadratic programming approach for mixed mode placement, *Proceedings of the 2005 International Symposium on Physical Design*, pp. 193–199, 2005.
43. http://er.cs.ucla.edu/benchmarks/ibm-place/

第6章　三维（3D）集成电路中的热通孔插入和热敏感布线

Sachin S. Sapatnekar

3D芯片的热挑战激发了能将热传输到热沉的片上热传导网络的需求。最显著的例子是热通孔组成的无源网络，其用于热传导而不必承担任何电气功能。本章从热通孔插入技术的总结开始。接着，强调3D布线问题，克服了诸多挑战，如传统2D布线扩展到第三维度和布线必须与热通孔竞争稀缺片上布线资源，特别是层间通孔资源。

6.1　介绍

三维集成技术将多个有源器件层打包在一起，允许在给定尺寸内增加集成层次。3D的优点很多，不仅包括减小互连长度和/或迟滞时间，还包括系统性能、功耗、可靠性和可移植性的提升。然而，3D设计也在如架构设计、热管理、电源传输和物理设计领域引入了相当大的挑战。

为实现3D系统的设计，必须开发从现今2D系统转移到3D拓扑结构的CAD工具。其中一方面是拓扑结构，因为线网可以在引入的第三维布线（或创建防止其他线网布线的阻塞框）。严格地说，3D技术不能实现第三维度上的彻底自由，因为允许的坐标与3D的层数相关，所以被限制到少量的可能性上。结果，在此情况下的物理设计常被称为相当于2.5D的问题。另一方面则与普遍的性能问题和特别的热问题相关。

这两方面使一个与2D IC设计流程大相径庭的3D设计/CAD流程成为必须。本书的第4章和第5章讨论了3D的布局规划和布局问题，本章特别关注有关通过互连的使用达到热平衡和布线的流程的问题。添加热通孔可以认为是在布线前或嵌入布线框架之中的后布局步骤。本章从将热通孔插入到3D电路以平衡3D系统中的温度分布的问题开始。随后，讨论同步布线和热通孔定位的方法，来更好地管理热通孔引起的阻塞框。

6.2　热通孔

3D电路中高温的潜在可能性有两个根源：第一，来自于单位面积内堆叠的更多的有源器件所造成的功耗增加；第二，来自从器件到封装以及外部环境的不适当

的热传导路径。

第一种来源可通过被充分研究的低功耗技术应对，第二种则需要提高从器件到封装的有效热导率来应对。硅是良好的热导体，它的热导率是典型金属热导率的一半甚至还多，但3D工艺中使用的许多材料都是强绝缘体。这些材料包括用于键合3D层的外延绑定材料，或场氧化物，或SOI工艺中的绝缘体。即使是在最好的分布了热源以控制片上温度的最优布局解的情况下，这样的热环境仍然严重限制了可移除热的总量。因此使用预先准备的金属作为散热通道，称为“热通孔”，这是总的热解决方案的重要因素。在没有热通孔的情况下，模拟显示3D芯片的片上峰值温度可达约150℃；这可以通过在布局后的后处理步骤中插入合理的热通孔来缓解。在理想3D工艺下，这些内部热通孔的尺寸在一侧是微米量级的。图6-1示出这样的通孔的实例。

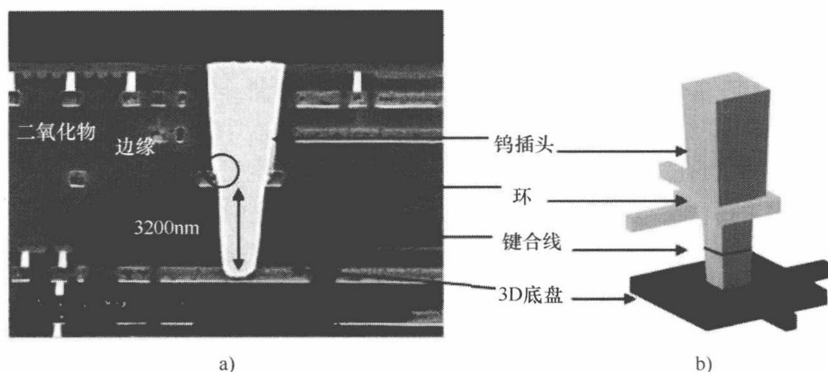


图6-1　3D层间通孔的　a）纵剖面SEM；b）等比例立体图[1]© 2006 IEEE

使用热通孔来优化热问题的思想早就在封装和印制电路板（PCB）的设计中采用了。特定的热通孔插入对2D芯片不必要，因为体硅就是很好的热导体。热通孔在3D领域里非常有吸引力是因为热流浓度大且邻接层在很多工艺中都被热绝缘材料分开。在此情景下，片上热通孔会在将热导向封装和热沉的过程中扮演重要角色，并降低片上温度。

在多芯片模块（MCM）领域中，Lee等人[2]研究了热通孔的排布并发现随着热通孔岛的增加，更多的热被导出，但也导致布线空间紧张。本章参考文献［3］研究了PCB和封装的设计参数和热通孔簇热阻的关系。将通孔簇简化为平行网络，使用热传导在垂直穿过厚度时比水平热扩散更有效的观察结果，即可确定这些关系。Pinjala等人实现了封装中热通孔的进一步热特征提取[4]。虽然这些文章的应用领域限于芯片内部热通孔的布局，但它们展示了热通孔的基本使用方法和特性。重要的是要注意到布线空间和热消除存在折中关系，表明热通孔应保守使用。热通孔可使用简化的热计算，热传导的方向主要沿热通孔的方向。Chiang等人建议在芯片衬底上使用另外设置的热通孔作为附加电气隔离通孔来减小有效热阻和潜在的热

问题[5]。其他几个更早的文献强调直接将热通孔集成到芯片内部来减轻内部的潜在的热问题[6,7]。因为多个电解质层的隔离效果，3D IC 相比 2D IC，热问题更严重并且热通孔能对其产生更大的影响。此外，互连结构能创建高效的热电路并有效降低芯片温度。

6.3 把热通孔插入到布局后的设计

在热通孔把热移到热沉和外部环境中起主要作用的同时，认识到这些通孔会引入对设计的限制也很重要。这些可以总结如下：

- 第一，每个通孔的接触 PAD 相当大，先进工艺下一般是微米量级。由于晶圆对齐后实现容易的可靠的连接的需要，这些定位上的限制增加。
- 第二，硅通孔在其周围形成机械应力，意味着在通孔周围存在电路结构的禁止区域。
- 第三，热通孔是布线的阻塞区域并将引发拥塞瓶颈。

为了管理这些约束，有必要加入设计规则以保证芯片的某些区域为放置热通孔预留，从而提供通孔、阻塞区和禁止区位置的可预测性。本章参考文献［8］使用了热通孔区域的概念，如图 6-2 所示，即置于两行单元中间：任何插入的通孔必须置于这些区域，尽管这些通孔不必都用到。这些布线阻塞区的密度在任一特定区域均受限，故能保证设计不会变得不可布线。

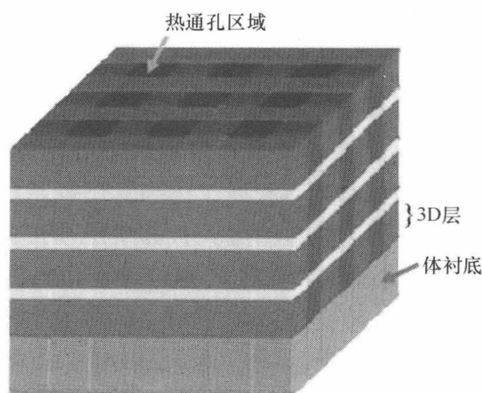


图 6-2 用于 3D IC 含热通孔区域的热网格

热导率的值（K）在任何特定方向均与排布在该方向的热通孔密度相关。对于所有实际目的，3D IC 中垂直热通孔的加入仅有助于 z 轴向热沉方向的热传导而水平方向热通孔对热传导的影响可忽略。

任何热优化必须联系到热分析。本章中，我们将使用第 3 章中介绍的技术重点介绍与此处讨论密切相关的细节。

从原理上看，放置热通孔的问题可视为在芯片中可放置热通孔的每一可能位置上确定两种热导率（对应有无金属）之一的问题。因此，实际上容易看出这样的方法将引入随可能位置数量以指数级增加的巨大的搜索空间。此外，以实践为出发点，执行全芯片的热分析，特别是在优化器的内部循环中以独立热通孔的粒度做热分析，是非常不合理的。在该细致层次，独立元素将必须对应到热通孔的数目，并且有限元分析（FEA）硬度矩阵将变得异常巨大。

幸运的是，有合理的方法克服这些问题。为了控制 FEA 硬度矩阵的大小，可以使用含元素数量相对较大的两层方案，其中每个区域的平均热导率可变。一旦该平均热导率选定，其将被转换回实现该平均热导率的元素中的热通孔的精确热分布。

本章参考文献［8］中的步骤使用了迭代方法进行热通孔插入来控制 3D IC 的温度。该步骤使用了基于有限元的热分析方法来计算片上温度并添加热通孔以改变芯片的热导率，从而降低 3D 堆叠中的温度。从初始配置开始，z 方向上的热导率逐次更新。

在每次迭代中，热通孔区域的热导率被修正，同时这些热导率反映了需要在此区域中应用的热通孔的密度。新的热导率由元素的 FEA 方程得到。在每次迭代中，特定元素的热导率通过加入热通孔实现微扰的引入。对于每个元素来说，该方法假定通过其的热流量在该微扰下保持不变，即

$$K_c^{\mathrm{old}} T_{\mathrm{old}} = K_c^{\mathrm{new}} T_{\mathrm{new}} \tag{6-1}$$

式中，T_q 和 K_c^{q}，$\mathrm{q} \in \{\mathrm{old}, \mathrm{new}\}$，分别对应元素硬度标记和元素角落的温度。基于对 8 节点长方体的硬度标志的数学分析，其可表示为

$$k_i^{\mathrm{old}} \Delta T_i^{\mathrm{old}} = k_i^{\mathrm{new}} \Delta T_i^{\mathrm{new}} \tag{6-2}$$

式中，$i \in \{x, y, z\}$，并沿指定方向 i；k_i^q，$q \in \{\mathrm{old}, \mathrm{new}\}$ 是在给定元素中通孔的有效热导率；$\Delta T_i^{\mathrm{new}}$ 为对应方向上的温度变化。定义热梯度如下

$$g_i^q = \frac{\Delta T_i^q}{d_i}, \quad i \in \{x, y, z\},\ q \in \{\mathrm{old}, \mathrm{new}\} \tag{6-3}$$

式中，d_i 是元素在 i 方向上的尺寸，其可化简为

$$k_i^{\mathrm{new}} = \frac{k_i^{\mathrm{old}} \Delta T_i^{\mathrm{old}}}{\Delta T_i^{\mathrm{new}}} = \frac{k_i^{\mathrm{old}} g_i^{\mathrm{old}}}{g_i^{\mathrm{new}}}, i \in \{x, y, z\} \tag{6-4}$$

该方法一个关键的观察结果是在控制温度的过程中温度梯度是最重要的指标。直观地说，中心思想就是如果一个区域具有高热梯度，那么加入热通孔将有助于改善整个热分布。实际上，上层一般比底层热，但在靠近热沉处，添加热通孔可降低底层的温度，同时能帮助减小版图中其他位置的温度。给定目标热梯度 g_{ideal} 和在前一个迭代的热梯度 g_i^{old}，就能在新的迭代中使用以下的计算更新：

$$g_i^{\mathrm{new}} = g_{ideal} \left(\frac{|g_i^{\mathrm{old}}|}{g_{ideal}} \right)^{\alpha}, i \in \{x, y, z\} \tag{6-5}$$

式中，$\alpha \in (0, 1)$ 是用户定义的参数，结合式（6-4）、式（6-5）得到

$$k_i^{new} = k_i^{old}\left(\frac{|g_i^{old}|}{g_{ideal}}\right)^{1-\alpha} \tag{6-6}$$

这在热梯度大于 g_{ideal} 时减小了 k 值并反之增大 k 值。在本章参考文献［8］中定义了对应的多种目标函数选择 g_{ideal} 的方式，如最大热梯度、平均热梯度、最大温度、平均温度、最大热通孔密度和平均热通孔密度。

一旦使用以上方法确定了热导率，下一步就是将这些转换到每个热区域的热通孔密度中。在一个热通孔区域中的热通孔量或金属化的百分比 m，也称为热通孔密度，可以由下式表示：

$$m = \frac{nA_{via}}{wh} \tag{6-7}$$

式中，n 为区域中独立热通孔的个数（显然受区域容量的上限制约）；A_{via} 是每个热通孔的纵切面面积；w 是该区域的宽度；h 是该区域的高度。热通孔百分比和垂直热导率的关系如下

$$K_z^{eff} = mK_{via} + (1-m)K_z^{layer} \tag{6-8}$$

式中，K_{via} 是通孔材料的热导率；K_z^{layer} 是该区域不含热通孔的热导率。使用该公式，对任何 K_z^{new} 热通孔的百分比 $K_z^{layer} \leqslant K_z^{new} \leqslant K_{via}$：

$$m = \frac{K_z^{new} - K_z^{layer}}{K_{via} - K_z^{layer}} \tag{6-9}$$

在每次迭代中，新的垂直热导率用来计算热通孔密度 m，以及每个热通孔的水平热导率。有效水平热导率 K_x^{new} 与 K_y^{new} 可通过以下计算求得：

$$K_{x[y]}^{new} = (1-\sqrt{m})K_{x[y]}^{layer} + \frac{\sqrt{m}}{\frac{1-\sqrt{m}}{K_{x[y]}^{layer}} + \frac{\sqrt{m}}{K_{via}}} \tag{6-10}$$

整个过程的虚拟代码见算法1。

算法1：热通孔布局（目标函数）

输入：在热通孔区域中热通孔布局的目标函数

输出：热通孔区域的热导率

将 g_{ideal} 赋给 g_{ave}

将 Ks 设为它们的最小值

计算温度分布

While 不收敛 *do*

Foreach 热通孔区域 *do*

$$k_z = k_z\left(\frac{|g_z|}{g_{ideal}}\right)^{1-\alpha}$$

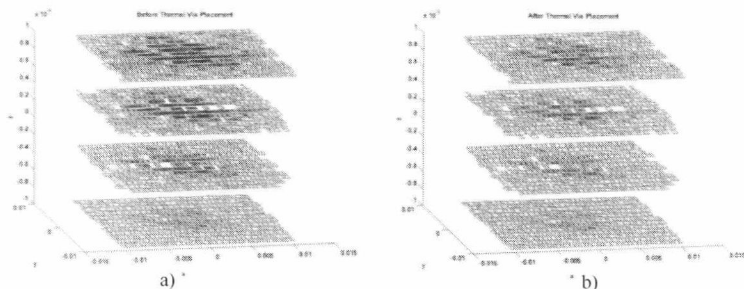
更新 K_x，K_y

（续）

计算温度分布
使用目标函数更新 g_{ideal}
计算热通孔密度

本章参考文献［8］中的技术已应用于一系列基准电路，其中超过 158000 标准单元。仅消耗几分钟时，插入的热通孔显示平均温度下降了约 30%。因此，热通孔插入对温度减小的影响远远超过了对热布局的影响。

图 6-3 示出了在热通孔插入前后，基准电路结构的 3D 版图。热图中黑色和亮色区域分别代表热和冷的区域。与大家猜测的不同，热通孔最集中的位置不在最热的区域。这背后的直观原因是：如果我们考虑最上层的中心，其热量是因为其下的层处在已提高的温度。因此，在第二层添加热通孔来去除热量可以有效且显著地减小顶层的温度。由此，热通孔插入最有效的区域是那些具有最高热梯度的区域。多个热目标的详细试验结果可参阅本章参考文献［10］。



a)　　b)

图 6-3　插入热通孔前后的结构热分布[9] © 2006 IEEE

本章参考文献［11］的工作展示了基于瞬态分析的热通孔插入方法。该方法探索了电—热二元性以及供电网格问题和热问题之间的关系。类似本章参考文献［12］中使用的总噪声违反矩阵，其取波形超过噪声阈值的量对于时间的积分，该方法使用了基于特定温度阈值的总热违反的积分。版图分片化为网格，优化约束为每个网格片中用于热通孔插入的现存空间的量和热通孔区域的总量，但其中并没有明确解释为什么两个约束都是必要的原因。使用模型降阶技术作为仿真引擎，使用顺序二次编程求解优化问题。

本章参考文献［13］中的后续工作使用电源网格来传导热并通过确定插入 TSV 的位置来优化电源网格，进而保证同时满足电压降约束和温度约束。如上述研究工作所述，版图分片化为小片，计算每片的通孔密度。

6.4　布线算法

一旦单元已经布局完毕并且热通孔的位置也确定，布线阶段就需要找到线网间

的最优互连。在 2D 布线中，优化线网长度、延迟和拥塞很重要。另外存在着一些关于 3D 特定的问题。首先，线网的延迟随其温度增加，导致更多的关键线网应该尽可能地避开最热的区域。第二，层间通孔是珍贵的资源，必须最优地分配给线网。第三，由于第三维度的加入，拥塞管理和阻塞框避开变得更加复杂。例如，一个穿过两器件层或多器件层的信号通孔或热通孔是布线必须绕开的阻塞框。

通过探索在线网的绑定盒子中确定准确布线的可行的灵活布线，以上所有问题均可得到解决。当线网长度的增加可改善延迟、拥塞或提供层间通孔指定的更多灵活性时，甚至可以考虑在绑定盒子之外进行迂回布线。

考虑如图 6-4 所示的三层工艺的布线问题。版图网格化为矩形方块，每个矩形方块具有水平和垂直容量。该水平容量决定了可通过该方块的线网数，而层间通孔容量决定了在该方块中可行的自由通孔数。这些容量不但是可分配给非信号线网（如电源和时钟线网）的资源也是热通孔使用的资源。如图中所示，对于一个单线网，可用的自由度包含在选择层间通孔位置的过程和选择每层中的精确路径的过程中。层间通孔的位置将依赖于在每个方格中资源的争夺情况。此外，关键线网需要尽可能避免高温方块。

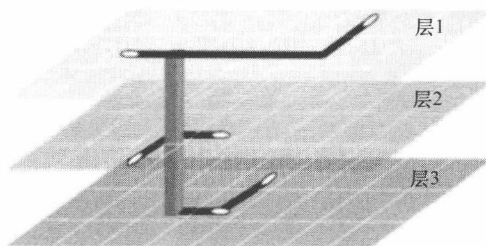


图 6-4 在三层 3D IC 工艺下对一个线网布线的示例[14] © 2005 IEEE

执行布线的基本网格图，在拓展到三维时，与标准的 2D 布线网格类似。每器件层都棋盘格化为 2D 网格，包含对应网格的顶点、相邻网格间的边，以及对应网格边缘容量的权重。相邻器件层顶点的互连对应这些位置可用的层间通孔百分比。

6.4.1 多层方式

本章参考文献［15］中的工作给出了带有热通孔插入的 3D 布线的初始化方法，随后在本章参考文献［16］中该方法改进得更加精细化。这两种方法取决于多层布线架构的顶部，与本章参考文献［17］类似。本多层架构的阶段包括递归粗化、初始解生成和逐层精细化，如图 6-5 所示。TSV 的分配首先在最粗的层次执行，而后在更精细的层次逐步执行。

本章参考文献［15］使用了本章参考文献［18］的紧凑热阻模型，即本质上与第 3 章使用的热阻模型相同。该方法的中心思想是通过两个步骤间迭代来确定热通孔和信号通孔的数量。在两层间给定网格中的热通孔分布使用简单的试探法，在

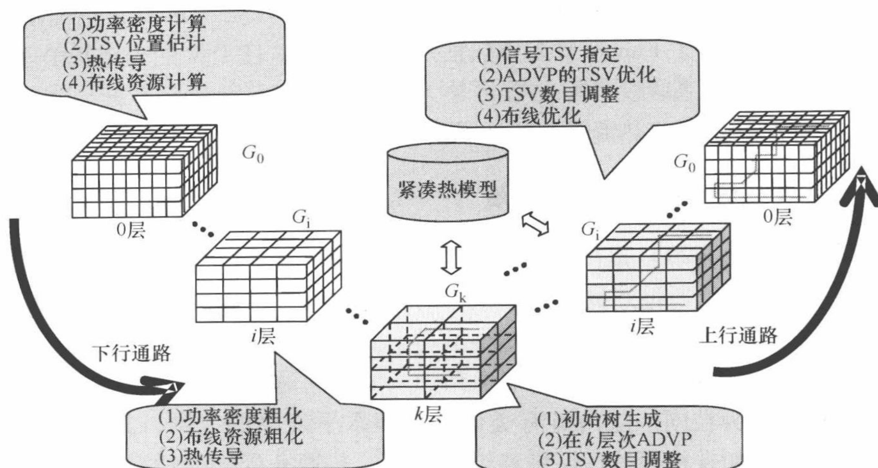


图6-5　包含TSV规划的多层布线框图[16] © 2005 IEEE

这过程中选择与这两个网格温度差成比例的数目。在多层布线中每个层次的信号通孔的插入都使用以下流程图描述的过程执行。

在多层方法的每层，即在多层结构的 $k-1$ 层，层间通孔规划问题将给定区域中的通孔指定到第 k 层的网格方块。该问题可建模为最小成本流向问题，其具有运输问题的形式。图6-6中的流程图的构成如下：

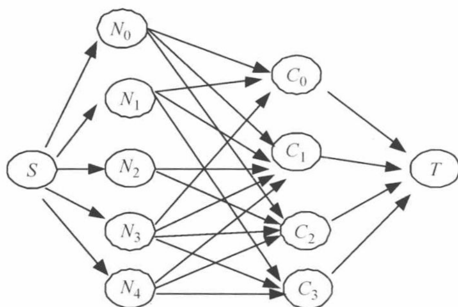


图6-6　对信号层间通孔规划的网络流程模型[15] © 2005 IEEE

1）流程图的源节点通过有向边连接到代表候选通孔的一系列节点 N_i；其边具有单位容量和零成本。

2）有向边连接第二套节点 C_j，从每个候选网格方块到热沉节点，具有与该方块能容纳的通孔数相同的容量，成本为零。容量使用考虑了方块和直接在其下器件层的块之间温度差的试探法计算得到（假定热直接向下流向热沉）。

3）热源［热沉］具有供应［需求］m，其等于在整个区域中的层间通孔数。

4）节点 N_i 通过有限容量的弧连接到方块 C_j，其成本等于将层间通孔 N_i 指定到方块 C_j 的估算的线网长度。

本章参考文献［16］中该工作的扩展也还是基于图6-5的多层布线架构。在该方法中，通孔规划方法得到了改进，通过使用称为替换方向TSV规划（ADVP）的方法。该方法也假定热流的主要方向为垂直方向。其引入了对TSV插入的非线性程序建模，但这确实太过昂贵，仅在文献中用作对照分析。提出的主引擎是迭代

的两步松弛化。首先，TSV 的位置（x，y）固定并且其 z 方向的分布是确定的。因此为该垂直方向开发了 Elmore 延迟类热估计模型[19]，并且 TSV 的分布基于理论计算的结果。然而，该结果假设 TSV 的数目是无限制的，与实际情况不符。随后，这些通孔根据每个方块的垂直热流量在器件层间水平移动。这两步迭代直到得出解。

6.4.2 使用线性编程的两段方法

本章参考文献［20］提出了一种在创建导热网络的同时满足拥塞约束的热敏感布线方法。该方法通过热通孔和热线网的合理插入，生成了不会出现热问题和布线容量违反的路径解，从而有效地减小了片上温度。像前面定义的，热通孔对应那些不具有任何电气功能但明确作为热通路加入的层间通孔。热线网则在本章参考文献［20］中定义为在同一层内具有类似功能但水平导热的线网，其在水平热传导上非常有用（例如在相邻器件层被绝缘层分开，热通孔受限的情况下）。热通孔执行传导到热沉的主要工作，而热线网帮助在多个热通孔间分布热路径。

图 6-7 示出了层间通孔如何减小相邻水平布线边的布线量。如果 $V_i \times V_i$ 个层间通孔通过网格单元 i 并且 $V_j \times V_j$ 个层间通孔通过相邻的网格单元 j，那么边 e_{ij} 的信号布线量将比初始的容量 C_e 有所减小，并且信号线网使用 W_e 需要满足

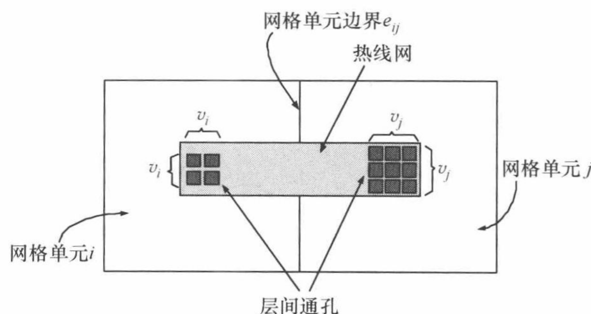


图 6-7 由 TSV 在相邻网格而减小的横向布线量；热线网集总并与热通孔形成散热网络[20] © 2006 IEEE

$$W_e \leqslant \min(C_e - v_i \cdot w, C_e - v_j \cdot w) \tag{6-11}$$

式中，w 为层间通孔的几何宽度。这里，两个减小的布线宽度中较小的一个定义为减小的边容量，这样就可以实现全局布线结果到详细布线解的灵活转换。另一方面，给定布线边的实际信号线网使用 W_e，式（6-11）也可以用来确定可通过相邻网格单元的层间通孔数目，从而不会在布线边出现溢出。由于减小温度需要插入大量的热通孔，为同时满足温度和可布线要求必须仔细规划。

一个简单的改善水平热传导的方式是识别出信号线网没有使用全部布线路径的布线边。剩余的布线通道就可以用热线网来连接临近网格单元中的热通孔。这些线网被直接连接到热通孔上以形成高效的热耗散网络，如图 6-7 所示。热线网使水平

方向的热传导成为可能并有助于垂直的热通孔有效地降低热点温度：对于仅可添加有限数目热通孔的热点，热线网可用于水平传导热量并通过临近网格中的热通孔去除热。热线网还有助于提供更一致的金属化，这对 CMP 抛光工艺[21]有益。

然而，将热线网或热通孔的延时添加到后布线后处理步骤，仅使用布线后未使用的剩余资源显然不是最优的。在理想情况下，这些应该在布线过程中分配。也就是说，由于热通孔和线网与信号线和通孔争夺布线资源，它们必须规划好来满足温度和可布线度的要求。本章参考文献［20］的方法提供了实现这一目标的方法。

本章参考文献［20］中的全局布线分为两段，其整体流程见图 6-8。算法的输入为给定电源分布棋盘化的 3D 电路。算法分两段进行，第一段根据流程图的前三块，第二段由迭代循环表示。实际上，观察结果表明该循环能在较少次的迭代后就收敛到优化解。

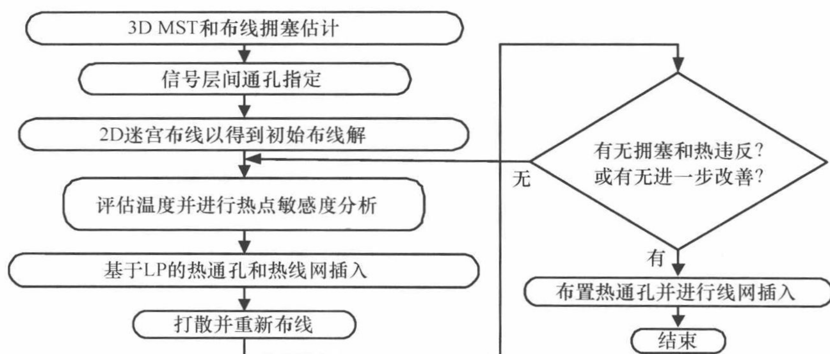


图 6-8　热敏感 3D 全局布线算法整体流程[20] © 2006 IEEE

第一段从最小生成树（MST）生成和布线拥塞估计步骤开始。随后，使用试探法网络基于流程的公式指定信号层间通孔。一旦这些层间通孔被指定，问题就缩小到每个器件层的 2D 问题。于是热驱动 2D 迷宫布线器用来分别解每器件层的问题。该布线器在标准迷宫布线成本函数中加入附加的温度项。

之后，第二段执行是包括向上跳重布线和基于 LP 的热通孔/线网插入的迭代布线过程。对于每个 $T_i > T_{target}$，$i=1, 2, \cdots, n$ 的 n 个温度违反热点，执行快速共轭灵敏度分析来找到 T_i 的灵敏度与在每个热通孔位置的热通孔数量的关系。如果灵敏度值超过一个位置的阈值，如 $S_{v,ij} \geqslant S_{th}$，那么该位置 j 就是热通孔插入的候选位置。与此类似，可定义候选热线网位置，并同时获得它们的灵敏度 $S_{w,ik}$。

使用基于该灵敏度的线性模型编写了一个用来插入热通孔和热线网的线性程序从而实现温度的小幅优化（与基于灵敏度模型的范围一致）。布线容量的小违反在此阶段是允许的，在认为它可在拆线重排步骤得到修正的理解的情况下。

令 $N_{v,j}$ 为在候选位置 j 插入的热通孔的数量，$N_{w,k}$ 为在候选位置 k 插入的热线网的数量，那么热点 i 的温度可降低 ΔT_i。LP 公式如下：

最小化
$$\sum_{j=1}^{p} N_{v,j} + \sum_{k=1}^{q} N_{w,k} + \Gamma \sum_{i=1}^{n} \delta_i \tag{6-12}$$

条件为：
$$\sum_{j=1}^{p} -S_{v,ij}N_{v,j} + \sum_{k=1}^{q} -S_{w,ik}N_{w,k} + \delta_i \geqslant \Delta T_i$$
$$i = 1,2,\cdots,n, \quad \Delta T_i = T_i - T_{目标} \tag{6-13}$$
$$N_{v,j} \leqslant \min((1+\beta)R_{v,j}, U_j - V_j), j=1,2,\cdots,p \tag{6-14}$$
$$N_{w,k} \leqslant (1+\beta)R_{w,k}, k=1,2,\cdots,q \tag{6-15}$$
$$\delta_i \geqslant 0, i=1,2,\cdots,n; N_{v,j} \geqslant 0, j=1,2,\cdots,p; N_{w,k} \geqslant 0, k=1,2,\cdots,q \tag{6-16}$$

最小化了总热通孔和热线网的目标函数与布线拥塞缩减的目标一致。为保证该问题可解，引入松弛化变量 δ_i，$i=1, 2, \cdots, n$。常数 Γ 在所有迭代中一致并选为足够大，以在热通孔和热线网资源在式（6-14）和式（6-15）的约束中足够按需要降低温度时压制 δ_i 的值到零。

式（6-13）需要在热点 i 的温度减小量加上松弛化变量 δ_i（引入以保证问题可解），在本次迭代中至少为 ΔT_i，其中 ΔT_i 为当前温度 T_i 和目标温度 T_{target} 的差值。

式（6-14）和式（6-15）分别与热通孔和热线网的容量约束相关。分别基于水平边界容量在同器件层的溢出和跨层的层间通孔容量的溢出。式（6-14）设置了热通孔插入数量的上限 $N_{v,j}$，含两个约束系数。$R_{v,j}$是可插入在 j 位置而不引发在相邻边上的水平布线溢出的最大附加热通孔数，可通过 $R_{v,j} = v_j - v_{cur.,j}$ 计算。其中 $v_{cur.,j}$是位置 j 的当前层间通孔使用数而 v_j 是可被插入在位置 j 并且不产生水平溢出的最大层间通孔数。在最敏感的位置添加更多的层间通孔对温度缩减的影响很大，因而，约束被有意地放大了 β 倍以临时允许该约束的违反，但这也将允许更好的温度缩减。这将潜在地导致热通孔指定后的水平布线溢出，但该溢出可通过向上跳重布线阶段解决。

对 $N_{v,j}$的第二个限制因素是总层间通孔的使用不能超过 U_j，其中 U_j 为位置 j 的层间通孔容量，并且约束公式取两个限制因素的最小值。类似地，式（6-15）考虑到水平布线溢出，设置了热线网插入数目的限制。$R_{w,k}$是不引发水平布线溢出的条件下在位置 k 能插入的最大附加热线网数。计算方法为 $R_{w,k} = m_k - m_{cur.,k}$，其中 $m_{cur.,k}$是位置 k 的当前热线网使用数，m_k 为位置 k 不引发水平溢出的最大热线网数。同样出于鼓励温度降低的考虑，$R_{w,k}$通过系数 β 松弛化，任何潜在的溢出将在向上跳重布线的布线阶段处理。

该方法的详细实验结果见本章参考文献 [20]。共生成了 4 套结果：①采用上述温度敏感布线算法的温度敏感布线（TA）；②使用上述一阶段方法的后插入布线（P），但随后将热通孔和热线网插入所有现存空间；③仅热通孔（V），其使用上述方法但仅使用热通孔而不使用热线网；④一致通孔插入（U），其使用与 TA 相同数目的热通孔和热线网但将其均匀分布在整个版图区域。图 6-9 中的试验结果显

示，与 TA 相比，P、V 和 U 都具有明显高得多的峰值温度。虽然 U 方案看起来具有与 TA 相似的热分布，但因这种情况导致了大量的布线溢出，所以不是合法的解决方案。TA 的线网长度成本仅比可视为非热敏感布线的 P 稍高。

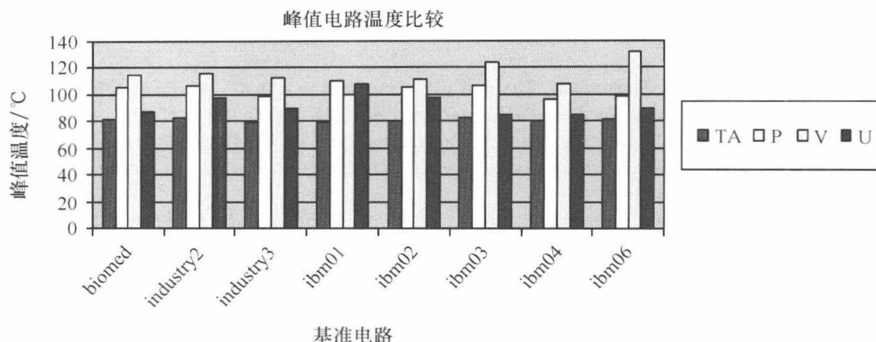


图 6-9 布线结果比较

6.5 结论

本章介绍了 3D IC 中布线和热通孔插入的多种方法。由于这两个问题竞争相同的有限片上互连资源，所以它们是相关的。这些布线资源的合理管理能提供热分布在保持可布线性基础上的显著改善。

致 谢

感谢 Brent Goplen 和 Tianpei Zhang，以及以 Jason Cong 为首的一批来自加州大学洛杉矶分校的小组，他们的工作对本章的内容做出了显著贡献。

参考文献

1. J. A. Burns, B. F. Aull, C. K. Chen, C. L. Keast, J. M. Knecht, V. Suntharalingam, K. Warner, P. W. Wyatt, and D. Yost. A wafer-scale 3-D circuit integration technology. *IEEE Transactions on Electron Devices*, 53(10):2507–2516, October 2006.
2. S. Lee, T.F. Lemczyk, and M. M. Yovanovich. Analysis of thermal vias in high density interconnect technology. In *Proceedings of the IEEE Annual Semiconductor Thermal Measurement and Management Symposium (Semi-Therm)*, pp. 55–61, 1992.
3. R.S. Li. Optimization of thermal via design parameters based on an analytical thermal resistance model. In *Proceedings of Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 475–480, 1998.
4. D. Pinjala, M.K. Iyer, Chow Seng Guan, and I.J. Rasiah. Thermal characterization of vias using compact models. In *Proceedings of the Electronics Packaging Technology Conference*, pp. 144–147, 2000.

5. T-Y Chiang, K. Banerjee, and K. C. Saraswat. Effect of via separation and low-k dielectric materials on the thermal characteristics of cu interconnects. In *IEEE International Electronic Devices Meeting*, pp. 261–264, 2000.
6. A. Rahman and R. Reif. Thermal analysis of three-dimensional (3-D) integrated circuits (ICs). In *Proceedings of the Interconnect Technology Conference*, pp. 157–159, 2001.
7. T-Y. Chiang, S.J. Souri, Chi On Chui, and K.C. Saraswat. Thermal analysis of heterogeneous 3D ICs with various integration scenarios. In *IEEE International Electronic Devices Meeting*, pp. 681–684, 2001.
8. B. Goplen and S. S. Sapatnekar. Thermal via placement in 3D ICs. In *Proceedings of the International Symposium on Physical Design*, pp. 167–174, 2005.
9. B. Goplen and S. S. Sapatnekar. Placement of thermal vias in 3-D ICs using various thermal objectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(4):692–709, April 2006.
10. B. Goplen. *Advanced Placement Techniques for Future VLSI Circuits*. PhD thesis, University of Minnesota, Minneapolis, MN, 2006.
11. H. Yu, Y. Shi, L. He, and T. Karnik. Thermal via allocation for 3D ICs considering temporally and spatially variant thermal power. In *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 156–161, 2006.
12. H. Su, S. R. Nassif, and S. S. Sapatnekar. An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts. In *Proceedings of the International Symposium on Physical Design*, pp. 68–73, 2002.
13. H. Yu, J. Ho, and L. He. Simultaneous power and thermal integrity driven via stapling in 3D ICs. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 802–808, 2006.
14. C. Ababei, Y. Feng, B. Goplen, H. Mogal, T. Zhang, K. Bazargan, and S. Sapatnekar. Placement and routing in 3D integrated circuits. *IEEE Design & Test*, 22(6):520–531, November–December 2005.
15. J. Cong and Y. Zhang. Thermal-driven multilevel routing for 3-D ICs. In *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 121–126, 2005.
16. J. Cong and Y. Zhang. Thermal via planning for 3-D ICs. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 745–752, 2005.
17. J. Cong, M. Xie, and Y. Zhang. An enhanced multilevel routing system. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 51–58, 2002.
18. P. Wilkerson, M. Furmanczyk, and M. Turowski. Compact thermal modeling analysis for 3D integrated circuits. In *Proceedings of the International Conference on Mixed Design of Integrated Circuits and Systems*, pp. 24–26 2004.
19. S. S. Sapatnekar. *Timing*. Springer, Boston, MA, 2004.
20. T. Zhang, Y. Zhan, and S. S. Sapatnekar. Temperature-aware routing in 3D ICs. In *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 309–314, 2006.
21. A. B. Kahng and K. Samadi. CMP fill synthesis: A survey of recent studies. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(1):3–19, January 2008.

第 7 章　三维（3D）微处理器设计

Gabriel H. Loh

3D 集成电路为计算机体系结构的创新发展提供了令人振奋的机遇。3D 集成电路技术可以应用到很多微处理器设计的潜在领域。本章我们将讨论一种设计方法，怎样重排一个传统的 2D 微处理器的模块，使其精确地摆放在多个晶圆层中。本章还讨论了不同的技术，以及折中方案，即面对芯片-芯片通信资源受限的情况，什么样的计算机架构可以有效地解决这一问题。3D 集成电路提供了许多方法来减少微处理器内的互连线，本章还讨论了更高层的设计方法，即怎样将互连线的减少转化成性能和功耗优势。

7.1　介绍

3D 集成电路为微处理器的设计（或重设计）提供了新的机遇。虽然本章的重点是高性能处理器，大部分的概念和技术可以应用到其他相关领域，如嵌入式处理器。本章的重点是通用的设计方法和 3D 处理器的架构。就未来的 3D 处理器而言，利用这种技术的最佳途径取决于许多因素，包括制造技术如何发展、散热和封装技术的进步、性能要求、功耗约束、工程局限以及其他问题。随着 3D 处理器架构的发展和成熟，本章中描述的技术为多种技术的组合。

本章内容按照年代顺序进行。我们首先探索 3D 处理器设计的简单模式，即堆叠较大的模块（如整个核），因此，只需要对传统的 2D 结构进行较小的改变。然后，我们考虑堆叠处理器的内部模块（例如，寄存器堆、ALU）为 3D 结构，这就具有了更大的灵活性和更高的优化途径。最后，我们研究更加深入的 3D 堆叠，即处理器内部的单个模块也可以打散，逻辑单元和互连线可以置于不同的层中。表 7-1 详细介绍了不同层次堆叠的得与失。

表 7-1　在不同的粒度 3D 堆叠的利弊概述

堆叠粒度	潜在益处	重设计需要努力
整个内核，缓存	新增的功能，更多的晶体管，混合工艺集成	低：重用现有的 2D 设计
功能单元块	减少的全局布线的延迟和功耗，降低了功耗的同时提高了性能	必须重新布局规划和重定时的路径
需要 3D 块级布局布线工具	可减少现存的 2D 块	逻辑门（块分裂）

（续）

堆叠粒度	潜在益处	重设计需要努力
全局、半全局和本地布线中减少的延迟/功耗	紧凑的块大小和重定义尺寸机会带来的进一步面积减小	需要新的3D电路设计、设计方法和版图工具 重用现有的2D标准单元库

未来在大众消费市场，量产3D集成电路的具体技术细节尚未知晓。但我们已经对什么技术可行，还是有把握的，但经济学、市场需求以及其他非技术因素可能影响这项技术的未来发展。也许是最重要的技术参数是芯片-芯片穿透硅的通孔尺寸大小。在很小的硅通孔间距下，处理器可以打散成更小的单元。如果硅通孔的间距很大，3D集成可能只能限制在模块级，甚至核级。在本章的其余部分，我们将重温硅通孔的尺寸是怎样影响设计的，并且在许多情况下，讨论如何能够绕过这些限制进行可行的设计。

7.2 堆叠完整模块

虽然3D微处理器将来最终可能实现堆叠更加微小的单元，如功能单元、互连线和逻辑门等，但3D集成方案的近期目标会简单得多。

从3D集成电路技术的出现到大规模制造工厂的建立，已经引起了一些显著的技术风险，因此在设计的其他领域的风险（即处理器架构）应尽量减少。考虑到这一点，对于3D堆叠的最简单的应用，应该是借用现有的2D设计方法。在本节中，我们将探讨3种通用的方法：扩大缓存层次结构、使用3D集成提供可选功能，以及系统级集成。

7.2.1 三维堆叠式缓存

避免任何3D集成中涉及复杂单元的重新设计（如完整的处理器流水线）。用最简单的方法集成更多晶体管的方式以增加更多的缓存和/或增加更多的内核。

即使是利用3D集成增加缓存容量这样一个简单的想法，对于构建3D堆叠结构的二级（L2）缓存仍然存在用于几种设计方案[1]。图7-1a说明了传统的双核处理器配备了4MB的二级缓存。由于二级缓存占用大约一半的芯片面积，堆叠硅第二层芯片相当于增加了额外的8MB缓存，总共有12MB的缓存，如图7-1b所示。注意，从底层二级控制器所在的芯片中心，到不同方向的最远的单元的距离大致相同。当硅通孔的延迟很小的时候，3D结构的变化对二级缓存的延迟几乎没有影响。相反，如果一个12MB的缓存利用传统的2D技术，如图7-1c所示，在最坏情况下，必然会增加布线距离，从而增加了缓存的延迟。

随着3D集成技术中晶圆堆叠或芯片堆叠方法的出现，各个堆叠的硅片层都是

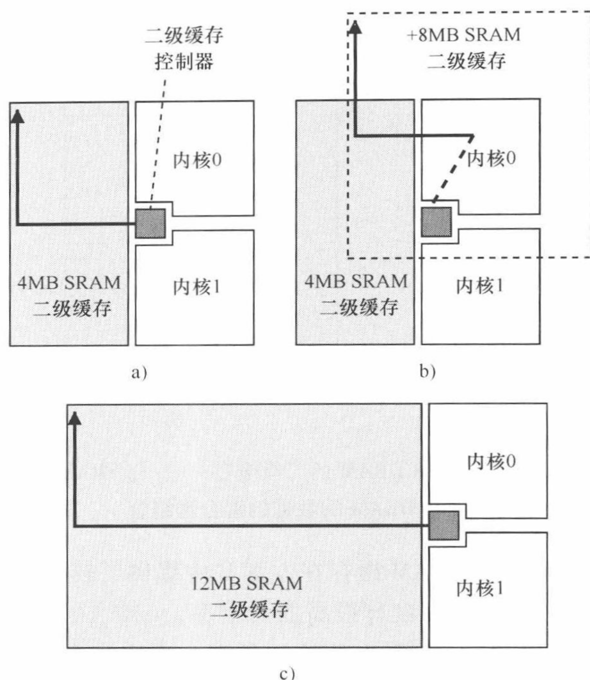


图7-1 a）常规2D与L2缓存的双核处理器；b）随着MB相同处理器的增加，即在3D组合中更多的二级缓存；c）对于支持整体12MB的二级缓存等效2D布局

在键合之前制造的。这就要求各个堆叠的硅片的制造工艺不必相同。另一种实现3D结构大容量缓存的方式是应用DRAM代替传统的逻辑/CMOS电路。基于DRAM的存储器比基于SRAM的存储器具有更大的存储密度（bits/cm^2），因此利用DRAM实现缓存，在相同的芯片面积上可以提供更大的存储容量。图7-2a示出了同一双核处理器中，基于SRAM的二级缓存已经被完全去除，取而代之的是堆叠在芯片顶端的基于DRAM的32MB二级缓存。

DRAM的堆叠设计意义主要是与SRAM相比可以提供更多的片上存储容量，但DRAM的访问延迟要比SRAM大得多。SRAM缓存具有10～20个周期的存取延迟，而DRAM缓存需要50～150周期（取决于行缓冲器的命中、预充电延迟和其他内存参数）。考虑表7-2所示的3个假设的应用。方案A工作集较小，适合应用4MB SRAM缓存。方案B有一个较大的工作集，不适合应用4MB SRAM缓存，但适合应用32MB DRAM缓存。方案C为存储访问模式，对DRAM和SRAM都具有很差的缓存命中概率。例如，对于方案A，DRAM和SRAM都具有很低的缺失率，但SRAM缓存较小的延迟需要更少的单次访问周期（CPMA）。方案B的工作集较大，SRAM缓存产生一个非常大的缺失率，导致了较高的CPMA。而DRAM缓存仍然比SRAM缓存具有更大的访问延迟，但与基于DRAM的片外内存相比，这仍然显著减

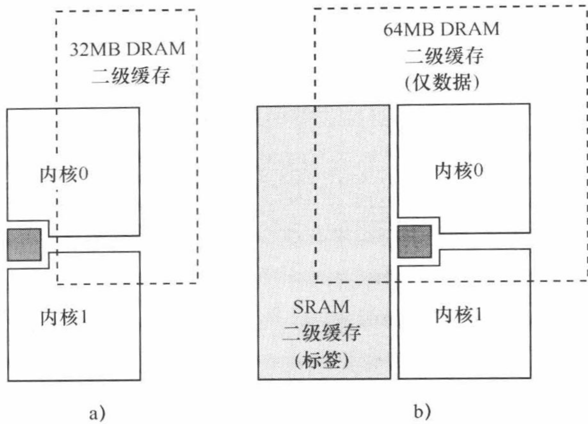


图 7-2 a) 堆叠 32MB DRAM 的二级缓存；b) 与 SRAM 标签和 3D 堆叠 DRAM 的数据的混合型组合

少了访问延迟。其结果是，DRAM 缓存在方案 B 中提供了较小的 CPMA。对于方案 C，SRAM 缓存和 DRAM 缓存都没有很高的命中率，CMPA 由决定缓存缺失率的访问延迟控制。此时，SRAM 具有更少的访问延迟，再次实现较低的 CPMA。3D 集成二级缓存的最佳情况取决于具体的目标应用，即处理器负载的工作量。

前面的例子表明，缓存的命中和缺失的延迟依赖于底层应用程序对存储器的访问模式。第 3 个方案结合了 SRAM 和 DRAM，设计了一个混合型缓存结构，如图 7-2b所示。底层芯片为 SRAM 阵列，只用来存储二级缓存的标签。顶层芯片使用 DRAM 来存储实际的缓存数据。在访问缓存的过程中，SRAM 标签可以迅速地提供命中/缺失指示。如果访问结果为缺失，经 SRAM 查找后，请求发送到片外存储器的控制器，访问片外存储器。这与由纯 DRAM 组成的存储器不同，纯 DRAM 组成的存储器不管是命中还是缺失，访问速度都较慢。在表 7-2 中的最后一行示出了这种混合 SRAM-标签/DRAM-数据设计与纯 DRAM 相比在执行三个不同程序时，是如何提高 CPMA 指标的。

表 7-2 对于不同缓存配置的访问延迟，3 个不同方案的命中和未命中的数量，假设一个 500 循环主存储器延迟每个存储器访问的平均周期（CPMA）。对于这个例子忽略这一级缓存

缓存组合	二级延迟		方案 A			方案 B			方案 C		
	命中	未命中	命中	未命中	CPMA	命中	未命中	CPMA	命中	未命中	CPMA
2D/4MB(SRAM)	16	16	900	100	6.6	200	800	41.6	100	900	46.6
3D/12MB(SRAM)	16	16	902	98	6.5	600	400	21.6	100	900	46.6
3D/32MB(DRAM)	100	100	904	96	14.8	880	120	16.0	100	900	55.0
3D/64MB(hybrid)	100	16	908	92	13.8	960	40	11.7	100	900	47.4

3D 堆叠利用短的，延迟较小的互连线，实现了末级大容量缓存直接放置于处理器核的顶部。另外，由于该互连接口并不需要 I/O 压焊块，不占用额外的版图面积，利用相对要小得多的硅通孔就可以实现一个很大的缓存接口。接口的预期宽度为缓存的数据宽度加上相关地址和控制位。例如，一个 64 字节宽的缓存，需要一个 512 位的数据总线加几十位的块物理地址和命令/控制信号。为了使信号传递更加容易，往往一个缓存需要建立两个独立的数据通路，用于通信的不同方向。虽然晶体管尺寸不断减少，但是硅通孔的尺寸和间距并不随之以同样的趋势减小。这导致硅通孔具有相对逐渐增加的尺寸和间距。要继续利用 3D 堆叠技术，硅通孔的参数需要不断地调整来适应缓存的不断变化的接口。例如，早期的 3D 堆叠缓存可能使用两个单向数据通路进行通信，但随着相对硅通孔尺寸的不断增加，人们可能需要使用单一的双向总线。另一个可能性是为了减少数据总线的宽度，应用流水线的方式在多个周期传输数据。这些例子说明设计是可以在很宽的范围内适应硅通孔的特性。

7.2.2 可选功能

随着 3D 集成技术应用中堆叠的芯片数量增多（总的层数），额外的压焊线制造工艺、成品率的下降等一些因素都会增加制造成本。此外，并非所有的市场需求都需要通过 3D 集成技术来提高集成度，从而增加产品的额外功能。有一种方法是利用 3D 集成技术作为一种手段来选择性地为处理器增加附加功能。例如，当一个 4 MB 的二级缓存可以满足大多数的市场需求，或是当缓存进行功能和性能的扩展所增加的成本和功耗将不适合某些应用（例如，成本低或移动设备），这种情况下，传统的 2D 单层微处理器是更可取的。但是在那些大容量的缓存可带来额外收益的领域（例如，服务器和工作站），3D 集成技术可用于提供缓存的扩展，而且不需要完全重新设计处理器。制造商可以利用 3D 集成技术，使一个单一的处理器设计满足广泛的应用。

7.2.2.1 自省型 3D 处理器

除了纯粹的性能增强外，3D 集成技术也可以用来为处理器提供新的功能。特别是 Loi 等人提出的 3D 结构自省型处理器[2]。程序员和工程师可以从访问有关微处理器的内部状态的详细动态信息中极大地获益。但是，现代硬件性能监控（HPM）系统的功能只允许用户监视有关处理器的一些基本统计信息，如缓存缺失的数量或分支预测命中率。有很多更丰富类型的数据对于软件和硬件开发人员也是非常有用的，可是要把这些数据集成到标准的处理器中，会显著地增加硬件成本。图 7-3a 描绘了一个概念上的处理器的布局规划图。在布局规划图中，每个点代表一个节点，在这里我们可以监控一些信息（例如，排序缓冲器占用率的统计、功能单元利用率及内存地址）。为了公开这些信息给用户，信息首先需要收集到的一些集中的 HPM 单元。用户通常可以配置 HPM 单元来选择所需的统计信息，额外的

硬件自省引擎可以植入来执行更复杂的分析，如数据分析、内存分析、安全检查等。图 7-3b 示出了处理器的整体布局规划，它受布线的影响。额外的互连线、中继器/缓冲器都需要分配额外的芯片面积。这进一步导致相邻功能单元距离的增加，从而增加了互连线的长度，引起性能下降。这样，整体的芯片尺寸可能会增大，从而增加芯片的成本。因此，虽然这种分析对于开发者非常有用，绝大多数用户却不会使用它。

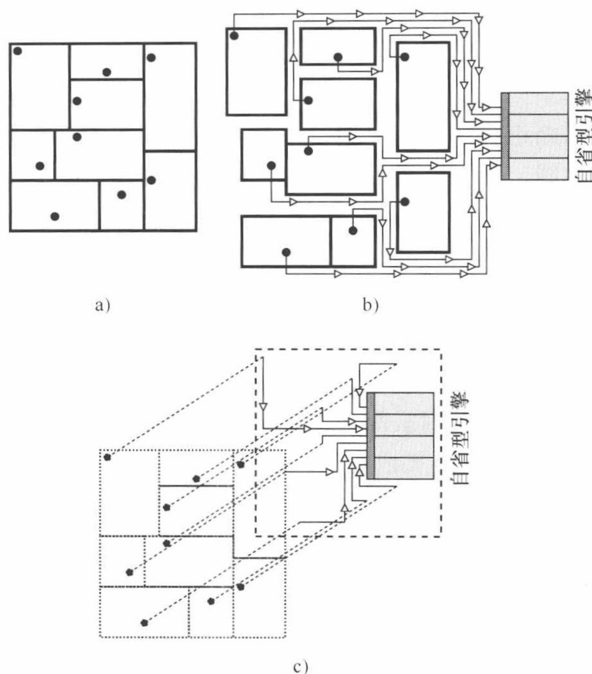


图 7-3　a）处理器与平面布局标记点的数据观测点；b）同一平面规划额外的布线空间和中继器转发数据的反省引擎；c）一个自省型 3D 处理器

自省型 3D 处理器的关键概念是以 2D 处理器为基础，尽量对 2D 处理器不做修改，以减少对已有大量商用处理器的影响，然后利用 3D 集成技术，有选择地为相对数量较少的硬件设计师、软件开发人员和 OEM（原始设备制造商）提供额外的分析支持。图 7-3c 示出了自省型 3D 芯片的两层结构。顶层给出了几个分析引擎的例子。它可以设计多种类型的自省层，然后为不同类型的开发者堆叠不同的引擎或设置不同的引擎。最主要的是，利用这种方法提供的功能添加自省单元对基本的处理器层的影响最小，处理器布局规划比较如图 7-3a、c 所示。

7.2.2.2　可靠的 3D 处理器

现代微处理器中器件尺寸不断减小，从而导致存储数据易被损坏，这是多种原因造成的，如更高的温度、电源噪声、互连串扰和高能粒子的随机影响（例如，α

颗粒）等。虽然目前处理器中的许多SRAM结构已经采用纠错码（ECC），以防止这些软错误[3]，但随着器件尺寸的进一步减小，未来处理器的脆弱性将不断增加。

假设常规的处理器可能会发生错误，并且可能产生不正确的结果，一种防止此种错误的方法是提供某种形式的冗余。复制两个相同的处理器，处理器被迫锁步运行。每个处理器产生的结果可以确认另一个。如果两个结果不相同，一方（或双方）必然有错误。这时，系统会刷新两条流水线，然后重新执行指令。如果应用三模冗余，以运算结果相同的多数作为正确结果，不需要重新执行指令，但必须复制多个该流水线显著增加了系统的成本。

除了使用锁步方式运行多个流水线，另一种方法是设计两条流水线分别作为前导执行内核和尾随检查内核。对于每个前导执行内核执行的指令，尾随内核会在稍后的时间重新执行此指令（不锁步），以检查出可能的错误。虽然这听起来与模块冗余的方法非常相似，这种架构能够优化尾随检查内核，从而降低成本。举例来说，可以不采用昂贵的分支预测器，而是尾随内核可以简单地使用由前导内核执行的计算结果。除了很少的软错误外，前导内核的结果绝大部分是正确的，因此尾随内核将从有效而完善的预测中受益。类似地，前导内核充当存储器预取器，使得尾随内核在缓存中几乎总是命中。还有许多其他的优化也减少了检查内核的成本，在此不作描述[4]。

即使有一个优化的检查内核，附加的流水线比原来未修改的处理器流水线仍然需要更多的面积。类似的动机内省型的3D处理器，并不是所有的用户需要其系统的可靠性处于这个水平，他们不希望因为他们不关心的功能、付更多的钱。3D堆叠结构也可用于有选择地为常规处理器增加检测内核，使其成为高可靠性的系统[5]。图7-4a示出了具有前导内核和检查内核的2D处理器架构。类似于图7-3，内核之间通信所需的额外的布线可能会增加面积开销。这种通信的延迟也因延迟了内核之间的交互信息影响了性能。

3D堆叠结构如图7-4b所示，避免了2D结构的许多缺点。首先，它允许检查内核可选，从而再不相关地细分市场，成本不会增加。第二，3D结构最小化了处于前导内核和检查内核之间的路由器的影响。这也影响了布线开销、基线处理器核的布局，以及核心之间通信的延迟时间。

检查内核比原来的前导内核需要更少的版图面积，主要是由于前面所述的各种优化。这种差异在面积配置文件中可能会留下一个显著的未使用的芯片面积。人们可以方便地使用这些面积设计额外的缓存。另一个可行的方法是使用基于上一代工艺技术（例如，65nm，而不是45nm）的堆叠层，如图7-4c所示。首先，基于上一代工艺技术生产的芯片更便宜。其次，在基于上一代工艺技术工艺的晶体管的特征尺寸都较大，从而使它们较少受软错误的影响。这样的做法有可能降低成本，同时提高了可靠性。

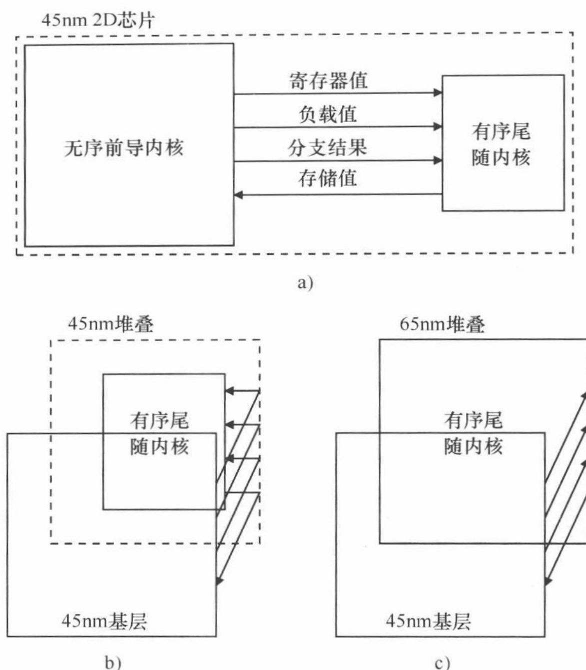


图 7-4 a）一个可靠处理器架构和一个小验证芯；b）3D 堆叠版本在尾随内核的顶部；c）和 b）相同，但尾随内核用较旧的工艺技术来实现

与常规的处理器相比，自省型 3D 处理器和 3D 堆叠结构的可靠性增强，只有两种可能的方式来使用 3D 集成提供可选的功能。这无疑有许多其他可能的应用，如堆叠结构的特定应用下的加速器以及可重构逻辑单元等。

7.2.2.3 硅通孔的要求

对于这两种内省型 3D 处理器和 3D 堆叠结构的可靠性检查内核，层间通信的要求不是非常高，也不会由硅通孔尺寸和间距限制。目前，晶圆键合技术可以提供每平方厘米成千上万个硅通孔（10000 ~ 100000）。在内省层所需的信号硅通孔的总数取决于需要收集和监控处理器层的分析引擎和数据量。跟踪使用率或各种架构的占用率只需要相对较小的计数器。为了降低硅通孔的要求，这些甚至可分割，使计数器有 k 位位于处理器层上。一旦 $2K$ 个事件完成，计数器需要传输进位到位于内省层的计数器的剩余部分（因此要求每个计数器只有一个硅通孔足以）。对于安全分析，该内省层将可能只需要检查内存访问，这将转化为监视几个内存地址总线和可能的 TLB 信息，共计不超过几百位。

3D 堆叠可靠性检查内核的主要通信需求是前导和检查内核之间通信的数据量。该峰值通信速率是由前导内核的提交率有效地限制。核之间典型的通信信息包括：寄存器结果、负荷值、分支结果和存储值。假定 128 位的数据值（例如，多媒体

寄存器），一个寄存器的结果、一个负载值、一个存储值、一个分支结果（包括方向和目标）要求小于512位。应用四路超标量处理器，这仍然只增加了2048位（或硅通孔）来实现前导和检查内核之间的通信。

7.2.3　系统级集成

先前3D集成的应用都集中在以某种方式扩展传统微处理器的功能。3D集成也可以用于集成微处理器以外的电路结构。例如系统存储器（DRAM）[6-9]、模拟电路[10]、闪存、图像传感器阵列，以及通常位于同一个系统主板上的其他组件。由于本章的重点是3D微处理器设计，我们不会进一步探讨这些系统级的问题。第9章提供了一个很好的说明，并讨论了一种可能的3D集成的服务器系统，称为PicoServer。

7.3　堆叠功能单元模块

在上一节中描述的几种可能的3D集成的应用，不需要任何实质性地改变微处理器的体系结构。对于最初几代的3D微处理器，尽可能地使设计采用微调的方法，以减少采用新技术相关的风险。3D集成将需要许多新的工艺、新的设计自动化工具、新的版图支持方法、新的验证和确认方法以及其他基础保证。以上这些最早的版本可能不能有效地支持复杂、精细地划分的3D结构。但是，随着技术的进步，计算机结构设计师可以通过新途径重构处理器流水线。

7.3.1　移除互连线

互连线延迟在现代处理器的设计中起着非常显著的作用。虽然每代工艺技术都提供了较前一代更快的晶体管，但互连线延迟的减小没有跟上同样的速度。因此，相对的互连线延迟随着时代的发展不断增加。逻辑门曾经是处理器周期时间的主要决定者，但现在互连线延迟已成为另一个主要的设计约束。图7-5a和图7-5b分别为在英特尔奔腾Ⅲ处理器的分支预测错误检测流水线和英特尔奔腾Ⅳ处理器[11]中数据经过的各级。由于具有更小的晶体管尺寸，奔腾Ⅳ流水线期望相对更高的目标时钟速度，但却具有相对较长的互连线延迟，奔腾Ⅳ流水线需要两倍多的级数。此外，有两个流水线级（在图中突出显示的），它们只是专用于驱动信号从芯片中的一个部分到另一部分。互连线的延迟变得很大，以致由该信号到达其目的地后没有剩余的时钟周期来执行任何有用的计算。但是，在一个3D结构中，流水线各级可以重组，先前较远模块现在垂直堆叠在彼此的顶部。因此，流水线各级所包含的互连线延迟现在可以完全移除，从而减小了流水线的整体长度。

一个旨在应对日益增加的流水线延迟的另一例子是Alpha 21264微处理器[12]。作为超标量处理器，它具有多个执行单元，需要一个旁路网络向所有的执行单元之

1	2	3	4	5	6	7	8	9	10
Fetch	Fetch	Decode	Decode	Decode	Rename	ROB Rd	Rdy/Sch	Dispatch	Exec

a) 奔腾Ⅲ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
TC Nxt IP		TC Fetch		Drive	Alloc	Rename		Queue		Schedule		Dispatch		RD Read		Exec	Flgs	BrCk	Drive

b) 奔腾Ⅳ

图 7-5　分支预测错误解析流水线英特尔奔腾Ⅲ和奔腾Ⅳ

间转发结果。这个旁路网络需要大量的布线，并且随着执行单元数目的增加，这些线的长度也增加[13]。如果是常规的处理器机构，旁路网络的延迟会严重降低 Alpha 21264 处理器的时钟频率。取而代之的是，Alpha 21264 的设计者把执行单元分成两组或两簇，如图 7-6a 所示。每个簇包含自己的旁路网络，从而在簇内部指令间零周期的转发结果。如果一个指令需要它的结果转发给其他簇的另一个指令，则该值必须通过二级旁路，从而带来额外的周期延迟。类似于奔腾Ⅳ额外流水线进程，这个额外进程中的旁路实际上仅会带来互连线延迟。然而在 3D 结构中，人们可以设法堆叠两个簇，一个位于另一个的顶部，如图 7-6b 所示，以消除长而慢的簇间布线，从而消除了因簇之间转发结果而产生的额外时钟周期。从缓存到执行单元路径，从寄存器堆到浮点单元路径[1]也都研究了使用 3D 结构转发簇间结果来消除额外的时钟周期延迟。

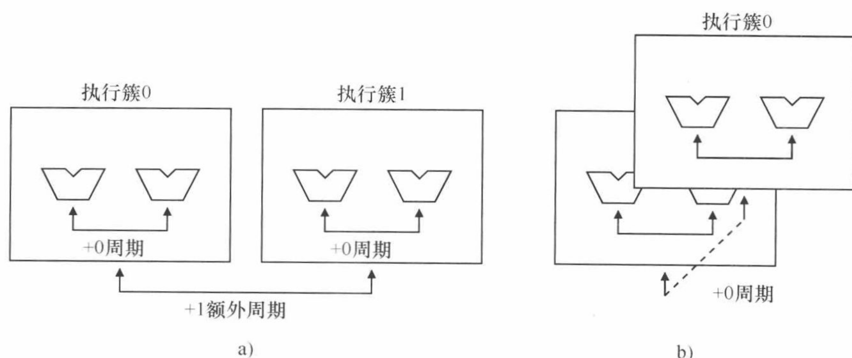


图 7-6　a) Alpha 21264 的旁路延迟执行簇，b) 一个可能的 3D 组合

这种方法采用 3D 集成结构来堆栈功能单元，设计不同的流水线元件，相比 7.2 节中讨论的粗放堆叠完整功能模块的方式具有很大的灵活性。这样做的好处是，可以缩短或是完全消除更多的块间布线，这反过来又可以提高性能并降低功耗。但当传统的微体系结构技术用于改进性能时，通常也需要增加功耗，或与此相反，任何降低功耗的尝试往往会导致性能损失。应用 3D 集成技术，我们在物理上

减少了系统中的布线的数量。减少的 RC 总线同时使延迟和功耗直接受益。

虽然有一些相关的成本，堆叠功能单元提供了更多的机会来优化处理器流水线。通过消除流水线额外进程，整体流水线组合可能变得更简单，但仍需要一些日常的设计工作，以修改流水线，然后验证新的设计依然如预期般工作。这说明，堆叠增加的成本超过了简单的重复使用一个完整的 2D 处理器内核的成本。请注意，每个功能单元的基本设计还是原有的 2D 设计。每个功能单元仅占据一个层。允许重用已有的宏单元库。在下一节中，我们将探讨设计方案，即使像寄存器堆和运算单元的基本模块也可以分割于几层中，但要花费更大的设计和工程的成本。

7.3.2 对硅通孔的要求

前述的技术是堆叠完整的模块（如内核，缓存）在彼此的顶部，此时需要较少的硅通孔，这是相对 3D 堆叠结构可以提供的硅通孔数量而言的。但是堆叠功能单元模块时，所需的硅通孔的数量视模块排列数量而不断增加。例如，在上一节中讨论的 Alpha 21264 的执行簇结构，要求各层之间的寄存器结果通过旁路。特别地，每个执行簇可以产生高达每循环两个 64 位的结果。这总共需要 4 个结果，这就增加了多达 256 位，再加上额外的用于物理寄存器标识符的位。此外，该存储器执行簇产生的每簇两个额外的结果也需要转发到两个执行簇。假设该存储器执行簇位于底部簇上，这增加了两个 64 位的结果即共 128 位。但总的来说，这仅仅增加了几百个硅通孔，只占两三个模块。如果 1 级缓存被堆叠在存储器执行簇的顶部，两个 64 位数据总线和 2 个 64 位地址总线共需要 256 个硅通孔。如果许多模块如此堆叠，每一个需要几百硅通孔，硅通孔总量会能很快攀升至数千个甚至上万个。

除了总的硅通孔数量的要求，局部通孔也可能会导致物理版图的问题，而导线长度也会引起后续的问题。考虑图 7-7a 中的两个模块，模块并行排列，16 条互连线连接它们。在这种情况下，对给定的硅通孔尺寸，层叠在彼此顶部的模块不会引起任何问题，如图 7-7b 所示。现在考虑图 7-7c 中的两个模块，这仍需要 16 条互连线，但这些块的总高度大为缩短。其结果是，没有足够的空间来容纳所有的硅通孔。在图 7-7d 中，所有硅通孔都短路在一起。利用不同的硅通孔布局，重排互连线来满足硅通孔间隔规则仍是可能的。图 7-7e 显示，一些局部的布线是潜在的解决方法。注意，局部布线会再次产生一些布线的开销，从而降低了 3D 集成结构减少互连线带来的益处。在极端情况下，如果硅通孔的要求非常高，面积非常有限，总局部布线可能完全抵消 3D 结构原有的减少互连线的好处。这些问题需要在放置模块开发的早期阶段，并与处理器数据和控制路径的整体布局中加以考虑。

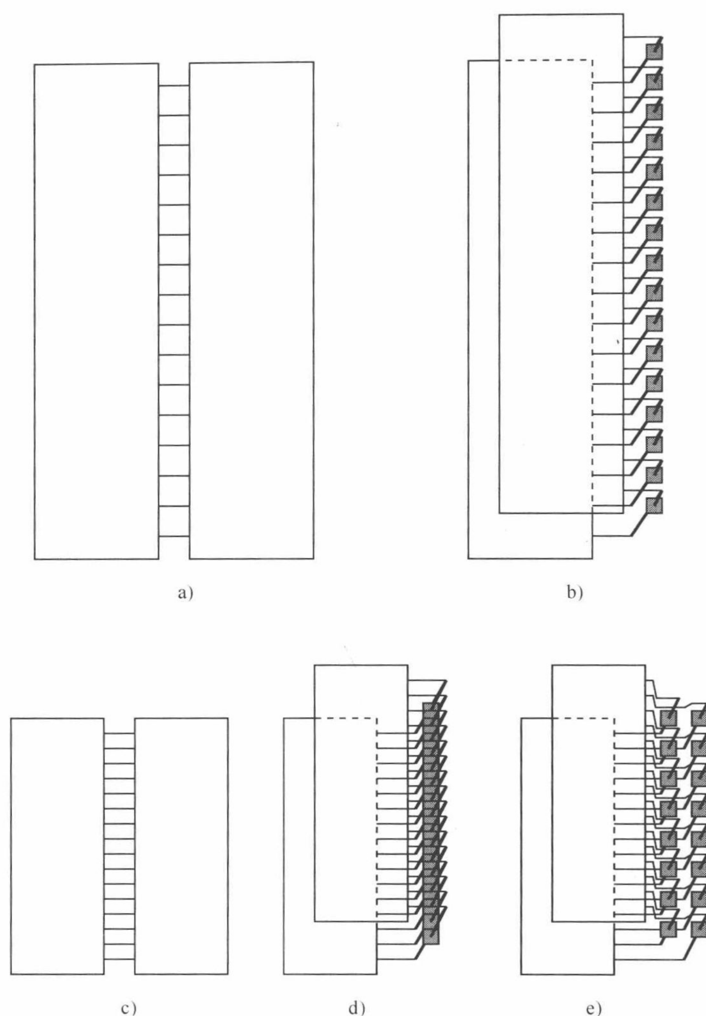


图 7-7 两个 16 个互连的可通信模块：a）原 2D 版本；b）3D 堆叠版本；c）紧密间距 2D 版本；d）不符合 TSV 的非功能性 3D 版本；e）可替代布局

7.3.3 设计局限问题

流水线设计为堆叠结构也可能会带来散热问题，这在第 4 ~6 章会有所讨论。再次使用 Alpha 21264 执行簇作为例子，堆叠一个执行簇在另一个的顶部可以降低关键路径互连线的长度，但也同时直接堆叠一个发热模块于另一个的顶部。由此带来的芯片温度的增加可能导致处理器更频繁地进入热保护机制。这反过来就导致较低的平均电压和时钟速度，从而使性能损失比旁路造成的额外周期增加还大。由于 3D 堆叠结构具用更大的设计灵活性，我们现在有更多的方式来打造更好的产品，

但也面临一个相对更大的设计局限，即同时平衡高性能、低功耗、低芯片温度、低重设计，及往往与这些设计要求相矛盾的产品设计目标和许多其他因素。

7.4 拆分功能单元模块

除了彼此顶部堆叠的功能单元的块，再往下细化，可以应用3D结构于实际的逻辑门，即可以分割单独的功能单元于多个层。现代高性能处理器一些关键模块关键路径的延迟都是互连线上的RC产生的。在这种情况下，重组功能单元块使其成为更紧凑的3D结构可以有助于减少模块内部的互连线长度，从而提高这些模块的工作频率。在本节中，我们只研究了两个微处理器模块，但这些技术和方法也可以通过扩展或改进来拆分其他模块。正在讨论的技术并不一定具体而详尽，但提供了一个出发点，以思考创造性的方式跨多个层设计实现电路。

7.4.1 三维缓存结构的折中

现代高性能微处理器的大部分芯片面积被多种缓冲存储器占据。许多市场状况较好的主流芯片，其二级/最后一级缓存已经占用了约一半的芯片面积。此外，处理器还有许多其他缓存，如一级缓存、页表缓存（TLB）、分支预测器。在本节中，我们将重点放在缓存上，但是，相同的想法可以很容易地应用到流水线中的基于SRAM存储器阵列。

我们首先回顾3D集成结构中粒度选择的影响。图7-8显示了几种不同的方法应用3D结构于二级缓存。在这个例子中，我们假设二级缓存已经被划分为8组。图7-8a说明了传统的2D布局。最坏情况下的布线距离取决于处理器内核的二级高速缓冲存储器存取逻辑的位置（用箭头示出），最大可达约 $2x+4y$，其中 x 和 y 是一个二级存储块的侧面的长度。图7-8b显示类似于第7.2节中描述的粗粒度的堆叠方式。请注意，虽然芯片的整体尺寸已经减少了一半，但在最坏情况下访问最远位时，线的距离相对于原来的2D情况并未改变。

当粒度达到更精细的水平，则二级缓存可以通过重新排列存储块实现完全堆叠。图7-8c显示了一个3D结构的存储块堆叠架构。假设每个处理器核也被拆分到两个层，利用第7.3节中讨论的堆叠的模块的方式。在这个例子中，最坏情况下的布线距离从处理器核到最远的位单元，此距离已减少 $2y$。该线长度减少可直接转化为在二级缓存访问延迟的减少。这种架构的一个优点是，各个存储块的版图基本保持不变，即使整个二级缓存已经跨过多个层，这种方法不要求缓存彻底的重新设计。

7.4.1.1 分割为3D结构的缓存

长的全局互连线是二级缓存访问延迟的主要来源，但每个存储块内部的互连线也极大地影响了整体延迟。长的全局互连线使用上层金属，这是典型的设计，以使

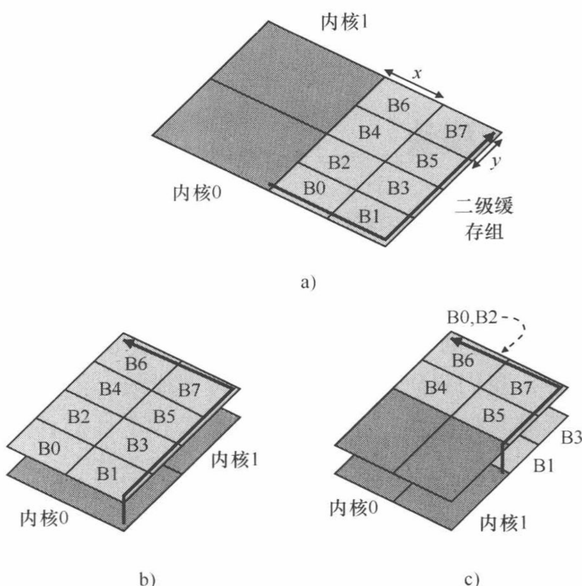


图 7-8 8 组二级缓存的双核处理器：a）2D 版本；b）在核芯上堆叠缓存组；c）组堆叠在组上

信号传输更长的距离。这包括谨慎选择导线的几何形状（例如，宽度与高度的比例）、考虑线间距，以及确定最佳位置和驱动的尺寸。模块内的互连线仍可能相对较长，但模块内的布线通常采用中间金属层，不利于长度优化。此外，模块内的逻辑通常显示出较高的密度，这就使置驱动于最佳位置和最佳大小几乎是不可能的。为了处理这个问题，我们也可以考虑拆分存储块于多个层。

虽然不同高速缓冲存储器的单元设计、集数、设置相关性、标签的大小和行大小等方面变化很大，但是基本的底层结构和电路拓扑结构上是大致相同的。图 7-9a说明了在读取单个位时，一个基本的 SRAM 组合结构。行译码器给出了被读数据的地址。行译码器置高一个且只有一个输出字线。被激活的字线使所有该行存储单元输出各自值于位线。列多路选择器由地址位控制，选择位线中的一个（正方一对）并传送这些信号到读出放大器。读出放大器可以通过快速检测一对位线之间的任何微小压差来加快读访问速度。对于一个传统的缓存，可能有多个平行的阵列来分别实现缓存数据和标签部分。另外，附加的多路选择机构，增强了标签比较逻辑，可以实现相连缓存。这样的逻辑结构不包含在本节中，本节只针对图中简单的结构。

有两种主要的方法可以把 SRAM 阵列拆分为 3D 结构[14,15]，我们将在下面依次讨论。我们首先考虑通过堆叠原有的列来拆分缓存，如图 7-9b 所示。总共有两种有效的方法来设计这种列堆叠电路。第一种，人们可以简单地把每一行视为分割到

两个层。这意味着原来的字线现在被劈开在两个层。这样，字线可更快地激活，因为现在字线长度大约只有原来的一半。而附加的缓冲器/驱动器可能需要进一步优化电路。输出列多路选择器也需要穿过两层进行分配。第二列堆叠结构是把它的列数翻倍，但每一行现在只有原来一半的单元，如图7-9c所示。这使字线是原来长度的一半（这与两个半长的字线相互连接不同），但它增加了行译码器的位数。

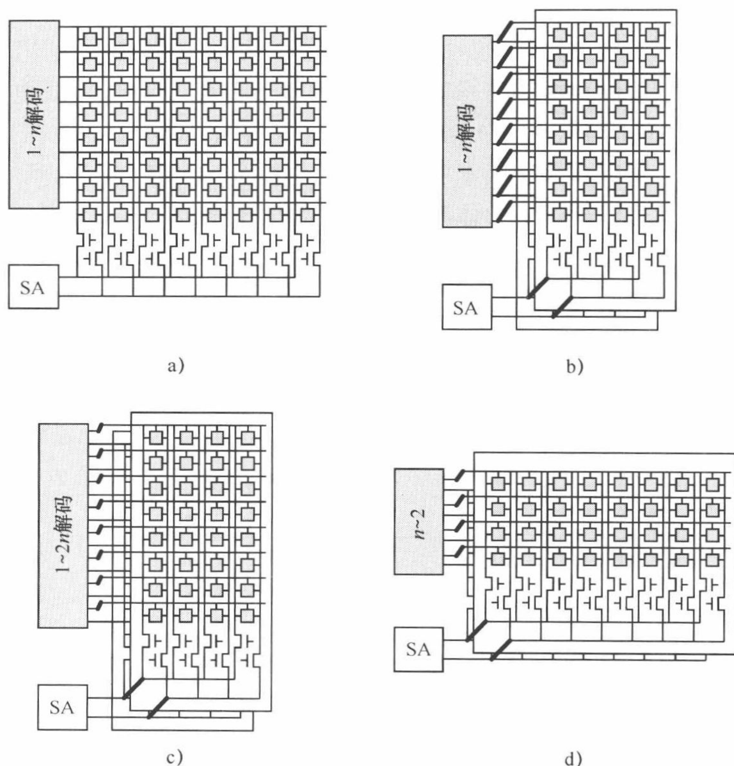


图7-9　SRAM阵列组合：a）初始2D布局；b）3D列对列布局与n行分割；c）3D列对列布局与$2n$个半长行；d）3D行对行布局

其他结构是行的堆叠，从而使SRAM阵列的高度大约减半，如图7-9d所示。这个结构需要拆分行译码器于两个层，所以选择单个行时或是从顶部或是从底部。因为只有一个单独的行被选择时，在列多路选择器工作之前，在理论上堆叠的位线可以共用。但由于延迟和功率的原因，位线通常分开作为单独的多路转换器输入端（见图7-9d），以隔离位线之间的电容。与标准的2D情况相比，这需要列多路选择器处理两倍的输入，但整个缓存中的读取延迟对列多路选择器延迟不太敏感，因为对多路转换器的控制输入端的设置过程可以与行译码和存储单元读取重叠进行。在这两个行堆叠和列堆叠的结构中，无论是字线或位线的长度都变短了。在任一情况下，此线长度减少可以同时转化为延迟和能量的减小。

现在，我们简要地介绍一些实验结果，以定量评估 3D 结构对缓存的影响。这些结果基于 65nm 器件模型的电路级仿真（SPICE）。表 7-3 示出了 2D 和 3D 结构不同大小缓存的延迟。3D 缓存由列堆叠结构组成。我们发现字线延时已经超过了位线延迟，因此堆叠的列（降低字线的长度）结构具有更快的缓存访问时间。总的趋势是，随着缓存容量的增大，3D 结构的相对收益（减少延迟%）也随之增加。这是因为，直观上，缓存越大，互连线越长。相对益处并不是单调增加的，因为不同大小的缓存都被分别优化，以使基本的 2D 结构具有最低的延迟。

当双层结构不能满足缓存要求时，缓存也可以分配至 4 个（或更多）层。表 7-3 的仿真结果还包括 4 层的 3D 结构。因为在大容量缓存中，字线的延迟往往超过了位线的延迟，所以在 4 层版本中，我们首先拆分 SRAM 阵列为列堆叠的结构。在该结构中，此时位线延迟对整体延迟的贡献要大于字线。因此，从两个层延伸到 4 个层，堆叠一半的行在另一半的顶部。仿真结果表明，此结构延迟进一步减小，但是 2 层增加到 4 层的相对收益要小于 1 层到 2 层的相对收益。

表 7-3　65nm 工艺下的模拟延迟结果（在 ns），实施各种 2D 和 3D 的 SRAM

缓存大小/KB	1 层 2D 延迟	2 层 3D 延迟	4 层 3D 延迟
32	0.752	0.635（-16%）	0.584（-22%）
64	1.232	0.885（-28%）	0.731（-41%）
128	1.716	1.381（-20%）	1.233（-28%）
256	2.732	1.929（-29%）	1.513（-45%）
512	3.663	2.864（-22%）	2.461（-33%）
1024	5.647	3.945（-30%）	3.066（-46%）

虽然列堆叠结构为两层 3D 缓存提供了最好的延迟改进率，但它不能有效地降低功耗。我们发现，当功耗是主要的设计参数时，行堆叠结构提供了更多的相对功耗收益。当字线访问 SRAM 中的某行时，该行中所有的存储单元都尝试连接它们各自的位线。而列多路选择器只选择其中的一个到灵敏放大器，但此行中所有单元的位线都充电/放电消耗了功耗。因此，通过行堆叠减少了位线的长度，这就直接降低了位单元的输出电容。从整条位线来看，堆叠后每条位线功耗都有所降低。与此相反，减少了字线长度节省的能量较少，因为行译码器每次读写时只激活一条字线。虽然这些结果可能由于缓存的结构不同而不同，但一般来说，3D 结构的确定主要取决于具体的设计约束和目标需要。设计一个 3D 电路，在最小化延迟，或能量，或面积情况下，都可能会有不同的最终结构。

7.4.1.2　处理硅通孔

本节中描述的 3D 结构缓存要求数量较多的硅通孔。例如，在列堆叠结构中，缓存需要两倍数量的字线。这可能是分拆字线的形式，如图 7-9b 所示，或两倍数量的原始字线，如图 7-9c 所示。在这两种情况下，所需的层间连接为每条字线一

条硅通孔。在理想情况下，所有这些硅通孔应放置在一列中，如图7-10a所示。但是，如果硅通孔的间距大于字线间距，可能会出现问题。图7-10b表明，较大的硅通孔的间距会导致硅通孔彼此重叠。

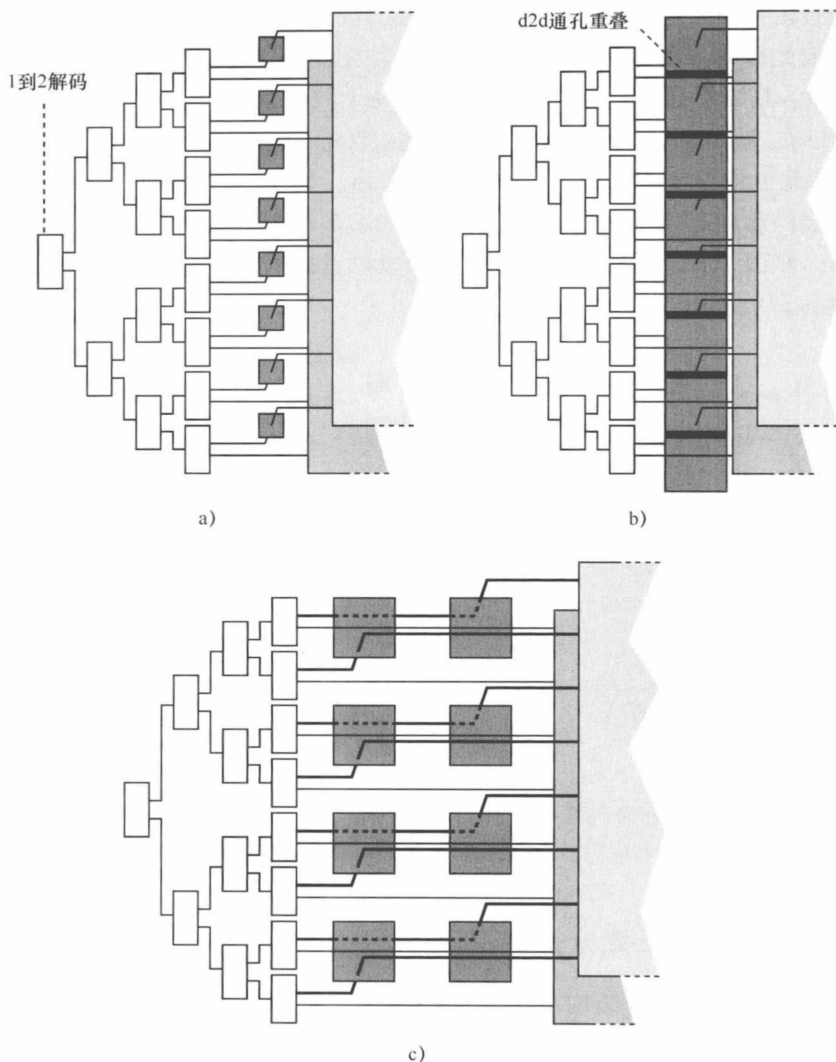


图7-10　对于列对列上3D的SRAM拓扑的行译码器的详情：a）足够小的TSV；b）太大的TSV；c）可替代布局容纳的大规模TSV

与2D芯片中用于金属层间连接的通孔放置类似，硅通孔可以重新定位以适应布局约束，如图7-10c所示。与硅通孔放置在模块之间用于通信的例子类似（第7.3.2节图7-7），这可能需要一些额外的层间布线，使信号从行译码器输出到硅通孔，然后再返回到原始的字线。只要该附加的布线开销（包括硅通孔）远小于由

3D 结构带来的布线减少，3D 缓存就将提供一个净延迟效益。在一个前面对后面的 3D 结构中，硅通孔必须穿过有源层，这会影响晶体管的布局。在这种情况下，需要附加空间分配给硅通孔，这又增加了缓存的整体尺寸。

当信号传输被限制（例如，当传输信号的数量超过可用硅通孔）时，信号和计算可以彼此折中。图 7-11a 示出了用于 16 位字线 SRAM 的行译码器线拆分/横跨 2 层的情形。其结果是，需要 16 个硅通孔（每字线 1 个）。图 7-11b 示出了另一种可能的布局，减少一半的硅通孔的字线的数量，但要复制额外的行译码器。图 7-11c借此更进一步降低了一半硅通孔的数量，但增加了更多的行译码器逻辑。这两种组合的总延迟将可能是近似的，但越来越多的逻辑译码器复制将导致更高的功耗成本。然而，合理地应用这样的方法给设计者提供了更多的技术选择来优化一个特定块的 3D 设计。

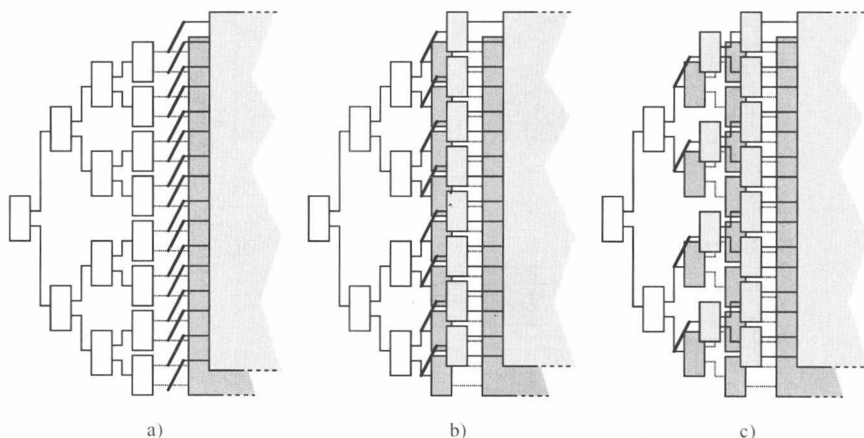


图 7-11　详细的行解码器：a）16 个 TSV；b）8 个 TSV，但一个额外水平的重复行解码器逻辑；c）4 个 TSV 具有两级重复逻辑

7.4.2　运算单元的三维分拆

缓存和其他的 SRAM 结构占用了现代高性能微处理器的绝大部分芯片面积，但微处理器中其他逻辑组件对性能也十分关键。SRAM 的结构是非常有规律的，并且分拆结构的不同策略也是直观的。对于含有更多的逻辑和更少规则的其他逻辑模块，分拆策略可能不那么直观。在本节中，我们将探索 3D 分拆运算单元的设计。我们特别专注于整数加法器，因为它们是逻辑和互连线的组合，具有一定的特定结构和规律，但规律性不如 SRAM 阵列。

7.4.3　三维加法器

虽然加法器有很多实施方式，在这部分我们只关注经典的超前进位加法器。许

多技术用于3D分割可以扩展或修改，以适应其他类型的计算单元，诸如乘法器和移位器。图7-12a显示了一个$N=16$位超前进位加法器的简单结构图。关键路径在于沿着从位［0］开始的进位传播生成逻辑，贯穿树形结构，然后到位［$n-1$］。

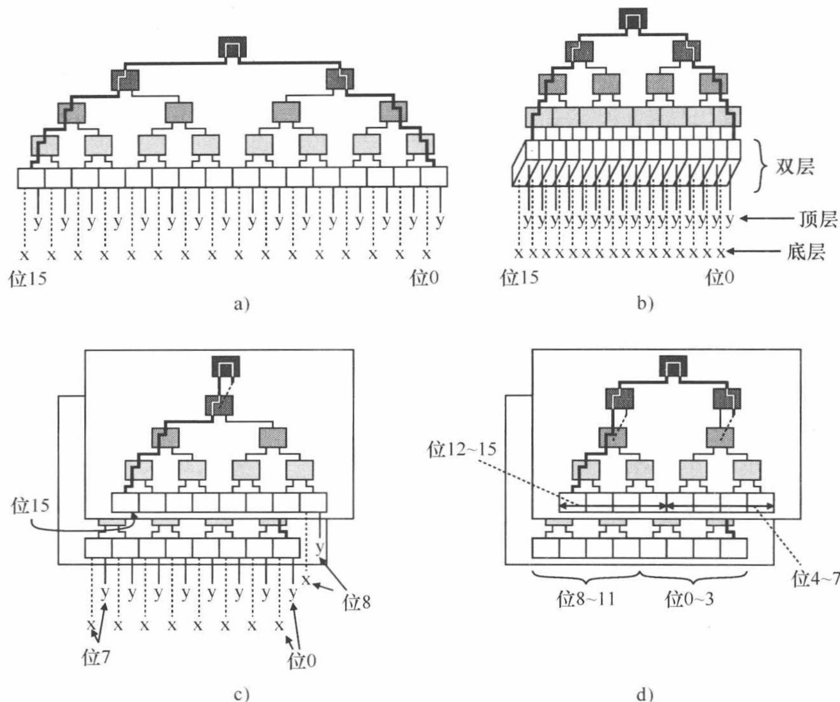


图7-12　a）两维先行进位加法器（LCA）的电路；b）3D LCA加法器与输入的分区；c）3D LCA显著度划分；d）混合显著度划分3D LCA

有几种自然的方式来划分加法器。图7-12b显示了一种实施方式，即根据输入来划分加法器。在这种方式下，输入x被放在底层，输入y放在顶层。这也需要第一级传播逻辑分开跨过两层，需要至少每位一个硅通孔。由于硅通孔和传播逻辑的大小和间距限制，加法器的整体宽度可能被减小。最好的情况下，如图中所示，会使沿着关键的进位传播生成路径的所有导线长度（在水平方向）减半。注意，逻辑的第一级之后，所有其余电路留在顶层上。

分拆加法器的第二种方法是根据重要性。我们可以将最不重要的位$\left(\text{即 } x\left[0:\frac{n}{2}-1\right]\right)$放在最底层，把最重要的位放在最顶层。图7-12c显示了这种方法的示意图。注意在初始的2D电路中最长的线（那些从根节点出去和进来的线）已经有效的被一条非常短的硅通孔替代，但是所有剩下的线长没有改变。注意和之前的输入分拆方法相比较，只有根节点需要来自2层的信号，因此总的硅通孔需求量根据输入n的大小决定。

还有许多其他可能的重排。图 7-12d 显示了一种显著的分拆方式的变形，其中低 n/2 位被放在电路的右侧，较高的位被放在电路的左侧。结果，一些中间的线被硅通孔代替，最后级的线长也已被减小。所有的 3 个 3D 组合方式能看作是相同的基本设计的不同实例，其中变化的参数是跨越两个层的树。在划分输入的方式中，在树的第一个级（在叶处）跨越两个层，而具有重要性的分区，它是跨越两个层的根节点。图 7-12d 的配置就像是其他两个的混合体：树的顶部的两级（在根部）在结构上相同于图 7-12b 的顶部，树的底部三级和图 7-12c 的底部非常相似。这样的布局用以支持 SIMD 运算，其中一个加法发生在右边，一个加法发生在左边。一个在物理上独立的位置定位逻辑加法的操作，可以决定由于功耗和门控所产生的互连线和控制带来的成本。

7.4.4 接口单元

最佳分拆功能单元的方式取决于设计目标，如最小化延迟、功耗或面积。最佳的分拆单元集合可包括组织组合，其中各个单元以局部亚最佳方式分割。考虑到在图 7-13a 中显示的 3 个相关单元块：一个寄存器、一个算法单元和一个数据缓存，通过它们可按位划分来分拆寄存器（最不显著的位在最低层），实现最低的延迟。输入划分（在不同层有不同的端口）数据缓存是缓存拆分的最佳配置，ALU 最受益于一种混合的组合结构，如之前结尾处描述的。这并不完全令人惊讶，因为每个块具有特性不同的关键路径，因此不同的技术可能是必要的，以获得最大的收益。

处理器由许多相互关联的模块组成，然而选择分开其中的一个模块，还可能对其他的模块有影响。考虑到相同的 3 个模块，其中从寄存器堆读出的数值转到加法器来计算一个地址，反过来提供给数据高速缓冲存储器，最终这些结果从存储器中得到并写回寄存器堆。图 7-13b 说明，这些模块用最小化每个单独模块的延迟的方式，每个模块已经拆分为两层。结果需要大量的硅通孔，因为每个模块的接口是不一样的。当数据从寄存器堆计算出，两个计算的最不重要的位在底层。然而，这个加法器需要位放在不同的层。加法器的输出反过来可能不能正确地对准数据缓存。数据缓存的输出最终可能需要使用旁路网络将结果直接转发给两个加法器和寄存器，因此需要更多的硅通孔来处理所有不同的接口。所有这些附加的硅通孔和绕线都增加了绕线开销，减小了通过 3D 结构带来的收益。其结果是，最佳整体配置可涉及简单地使用根据重要性拆分（例如）所有的组件，如图 7-13c 所示。而这意味着，在局部我们使用亚最佳 3D 结构的加法器和数据缓存仍可获得全局最优配置。

一个块 3D 拆分策略的选择对于在处理器的其他许多块可以有深远的影响。任何根据重要性拆分数据路径的方式很可能迫使所有其他数据路径以相同的方式拆分。指令高速缓冲存储器的 3D 结构选择可能反过来限制解码逻辑的布局。例如，如果指令高速缓冲存储器传送其指令的一半到底层，余下的指令传递到顶层，则译

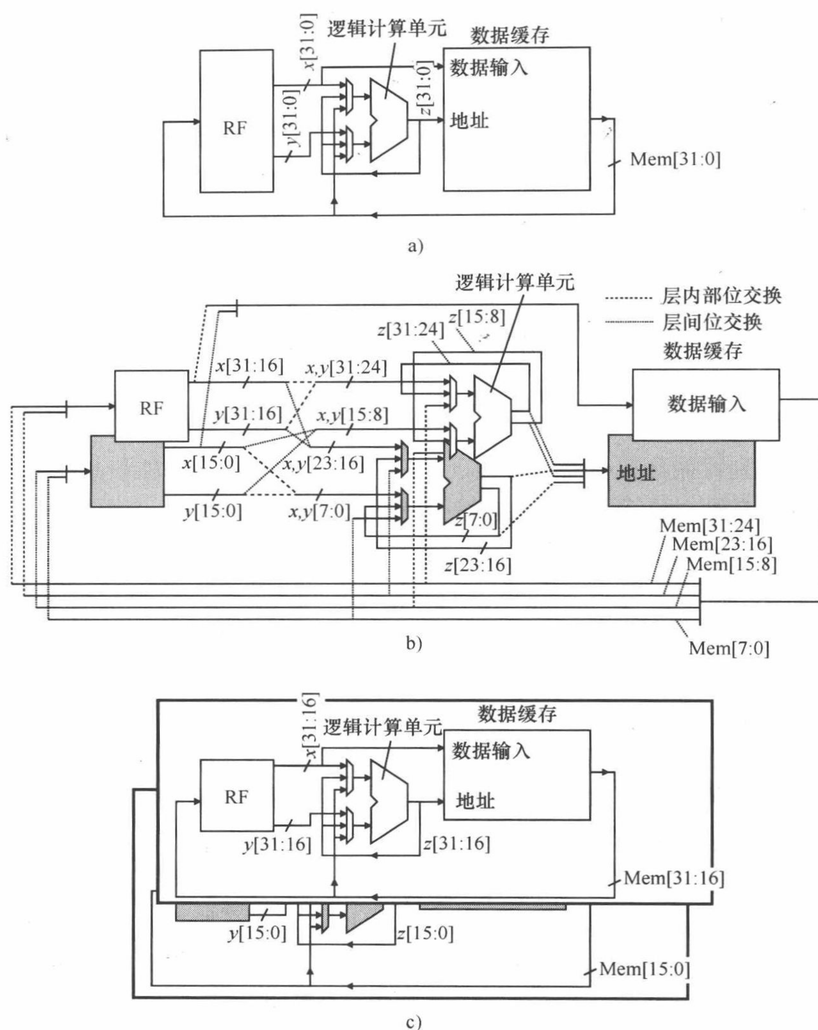


图7-13　a）2D寄存器堆（RF）、逻辑计算单元（ALU）和具有数据通路和旁路的一个数据缓存；b）每个单元使用不同的划分方法的3D结构；c）所有单元使用相同拆分的3D结构

码逻辑将类似地将一半分配于每个层中。这些都不是硬约束，但不同接口的成本来源于使互连线和信号匹配的附加硅通孔。

7.5　结论

从计算机架构师的角度来看，3D集成提供了两个主要的好处。第一，3D空间物理结构能显著降低导线长度。第二，不同制造技术的器件可以一种3D堆叠的方式紧密集成并结合。

像“3D 减小线长”的简单陈述有许多不同的解释方式，可用于微处理器设计。计算机架构师怎样利用互连线长度的减少？之前的部分已经讨论了 3D 集成处理器设计的具体技术，我们现在讨论一些更高级别的影响。首先，和前面部分的技术并不一定是相互排斥的。例如，一种可能的选择来堆叠某些块在其他块的顶部同时分割一些单元跨越多个层，然后在第 3 层集成一个完整末级缓存。处理器的不同模块具有不同的设计目标和约束条件，从而需要不同的 3D 设计策略，以提供最佳的解决方案。

从整体结构的角度来看，对于减少导线也提出了几种不同的选择。正如之前部分的讨论，线的减少和再布局能使整个流水线级减少。在其他的情况下，重要的线延迟的两个阶段可以合并为一个阶段。除了由于更短的流水线使得性能提高，还可以得到整体复杂度的减小。例如，深度执行流水线可能需要多层次的结果旁路，使相关的指令背到背循环执行而没有拖延。延迟时间、面积、功耗和其他常规旁路设计相关联的复杂性对许多相关的参数具有巨大影响[13]。这样，消除一个或多级的旁路能大幅度减小复杂度。

流水线级的减小有许多显著的好处，例如性能的提升、流水线控制逻辑的减小和功耗的降低。然而，这些变化可能使得进一步整体提升流水线的架构成为可能。例如，有许多缓存器和队列在现代超标量处理器中，有几个周期的缓冲指令以容忍各种流水线延迟。通过减少整体的流水线长度，减小一些指令的大小是可能的，或者在某些情况下甚至可以完全地删除它们。有许多微架构技术被设计来容忍互连线的延时，但是如果 3D 集成极大地减小这些关键路径的影响，流水线的整体架构将可能为功率、面积和复杂度等提供收益。

代替消除流水线阶段，3D 也可能被用来减少每个流水线阶段的时间消耗。这导致更高的时钟频率和因此提高了性能，虽然可能有更高的功率消耗。注意这区别于传统增加时钟速度的流水线技术。传统上，增加处理器的频率需要次驱动流水线到大量的短的（低延时的）阶段。用 3D 技术，阶段的总数量可能被保持常量同时减少每个阶段的延时。在床柜的流水线，这种架构简单地将固定工作量和休息变为更小的片段，而 3D 技术实际的减少了总的工作量通过移除线的延时。

另外的潜在的选择是不使用减少互连线的方法来提高性能而是将时间裕度转化为功耗的减少。例如，关键路径的门和驱动器经常使用尺寸更大的晶体管驱动电容来增加速度。如果去除这条路径可以减少电路的延时，那么电路的设计者可以减少晶体管的大小，这样反过来减少他们的功率消耗。这为完全的改变设计提供了机会，从快速的动态的/多米诺逻辑转化为低功率的 CMOS 门设计。在其他的单元块中，晶体管可具有更长的沟道，这使得它们更慢，但减少了它们的漏电流。

更早的时候，我们讨论了不同的 3D 类型（例如堆叠或者分拆）如何结合到一个系统中，用不同的方式优化不同的模块。以类似的方式，3D 技术可能以不同的方式应用到不同的处理器模块来优化不同的设计目标，例如时序、面积或者功耗。

虽然我们已经极大地关注了线的延时、性能和功耗等问题，平衡设计复杂度和新器件的再设计、测试和验证的成本也是十分重要的。在一些情况下，精细划分的3D模块可以提供更大的好处，但是工程上的成本和对整体项目计划的影响和风险，可能迫使设计者使用更加保守的结构。

在本章中，我们探讨3D集成在几个不同级别的应用，但是没有直接地试图回答一个精确的3D架构是什么这个问题。这时，我们只能猜测答案，而最佳答案将取决于许多未知因素。正如本章讨论多次的，结构的确切组合将在很大程度上依赖于制造工艺中所提供的硅通孔的精确尺寸和间距。随着冷却技术的未来改进，计算机架构师也许能够采取更紧密的结构，通过减少互连线解决问题而不是专注于热管理问题。如果冷却技术没有迅速进步，那么最佳的3D设计可能看起来非常不同，因为架构师必须更仔细地管理处理器的功耗密度。

致　谢

通过数年与许多研究人员合作，本章中提出的很多工作和思想得到发展。特别是曾在英特尔公司工作过的Bryan Black和其他研究人员，宾夕法尼亚州立大学的Yuan Xie和佐治亚理工学院的Kiran Puttaswamy。该研究的资金和设备也由美国国家科学基金会、英特尔公司，以及由半导体研究公司的焦点中心研究项目资助下的电路和系统解决方案中心（C2S2）共同资助。

参考文献

1. B. Black, M. Annavaram, E. Brekelbaum, J. DeVale, L. Jiang, G. Loh, D. McCauley, P. Morrow, D. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. P. Shen, and C. Webb. Die-stacking (3D) microarchitecture, *International Symposium on Microarchitecture*, pp. 469–479, 2006.
2. S. Mysore, B. Agarwal, N. Srivastava, S.-C. Lin, K. Banerjee, T. Sherwood. Introspective 3D chips, *Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 264–273, 2006.
3. C. McNairy, R. Bhatia. Montecito: a dual-core, dual-thread Itanium processor, *IEEE Micro*, 25(2):10–20, 2005.
4. T. Austin. DIVA: A dynamic approach to microprocessor verification, *Journal of Instruction Level Parallelism*, 2:1–26, 2000.
5. N. Madan, R. Balasubramonian. Leveraging 3D technology for improved reliability, *International Symposium on Microarchitecture*, pp. 223–235, 2007.
6. G. Loh. 3D-stacked memory architectures for multi-core processors, *International Symposium on Computer Architecture*, pp. 453–464, 2008.
7. C. Liu, I. Ganusov, M. Burtscher, S. Tiwari. Bridging the processor-memory performance gap with 3D IC technology, *IEEE Design and Test*, 22(6):556–564, 2005.
8. G. L. Loi, B. Agarwal, N. Srivastava, S.-C. Lin, T. Sherwood. A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy, *Design Automation Conference*, pp. 991–996, 2006.

9. T. Kgil, S. D'Souza, A. G. Saidi, N. Binkert, R. Dreslinksi, S. Reinhardt, K. Flautner, T. Mudge. PicoServer: using 3D stacking technology to enable a compact energy efficient chip multiprocessor, *Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 117–128, 2006.
10. G. Schrom, P. Hazucha, J.-H. Hahn, V. Kursun, D. Gardner, S. Narendra, T. Karnik, V. De. Feasibility of monolithic and 3D-stacked DC-DC converters for microprocessors in 90 nm technology generation, *International Symposium on Low-Power Electronics and Design*, pp. 263–268, 2004.
11. G. Hinton, D. Sager, M. Upton, D. Boggs, D. Carmean, A. Kyler, P. Roussel. The microarchitecture of the Pentium 4 processor, *Intel Technology Journal*, Q1, 2001.
12. R. Kessler. The Alpha 21264 microprocessor, *IEEE Micro*, 19(2):24–36, 1999.
13. S. Palacharla. *Complexity-Effective Superscalar Processors*. PhD thesis, University of Wisconsin at Madison, 1998.
14. K. Puttaswamy, G. Loh. Implementing caches in a 3D technology for high performance processors, *International Conference on Computer Design*, pp. 525–532, 2005.
15. Y.-F. Tsai, Y. Xie, N. Vijaykrishnan, M. J. Irwin. Three-dimensional cache design using 3DCacti, *International Conference on Computer Design*, pp. 519–524, 2005.

第8章 三维（3D）片上网络架构

Yuan Xie　Narayanan Vijaykrishnan　Chita Das

片上互连被预测是设计具有多个同质或异质处理器核和功能模块的多处理器多核芯片（CMP）和片上系统（SoC）架构的基础性问题。为减轻互连危机，一个可靠的选择是片上网络（NoC），其中一个通用片上互连网络使用开关矩阵连接 IP 核或处理单元，代替了传统定制全局片上互连。这样基于包的通信网络由于其可规划性已经被广泛接受并被认为是未来 CMP 的 SoC 的有效解决方案。在本章中，我们研究三维集成电路和 NoC 的两者组合，因为它们都是为减轻互连规模造成的挑战而设计的。本章将从片上网络架构的简要介绍开始，随后讨论多种网络拓扑结构在 3D NoC 设计中的设计空间，同时探讨 3D 片上路由器设计的多种技术。最后，介绍一个带有堆叠在多核 CMP 上的存储器的 3D NoC 设计实例。

8.1 介绍

随着工艺尺寸的缩小，使在芯片上集成 10 亿个晶体管成为现实。例如，最新的 Intel Xeon 处理器包括 23 亿个晶体管[25]。这样的集成层次必须使用并行机制来有效利用晶体管。于是，现代超线程处理器引入了许多复杂的微结构特性，如多指令处理、动态规划、乱序执行、投机执行和动态分支预测[26]。然而，为了保持性能的增长，未来的超线程微处理器甚至将依赖更复杂的架构创新。电路限制和受限的指令级并行机制将削弱通过增加架构复杂度的超线程模型[26]带来的益处。增加的宽度造成队列尺寸和寄存器堆复杂度平方级的增长。此外，随着执行单元数目的增加，连线和互连逻辑复杂度开始反过来影响性能。这些问题导致多处理器芯片（CMP）作为一种可行的替代复杂超线程架构的方案出现。CMP 是由简单、紧凑处理核组成的非中心式的微架构，其随集成度的增加可更有效地规划。9 核单元处理器[6]、8 核 Sun UltraSPARC T1 处理器[13]、8 核 Intel Xeon 处理器[25]，以及 64 核 TILEPro64 嵌入式处理器[1]都标志着这类系统不断增长的普及程度。

在设计具有多个同质或异质核和功能块的多核多处理器芯片（CMP）架构中，一个基本问题是片上互连矩阵的设计。如第 1 章所讨论的一样，片上互连被认为将是在工艺进一步缩小到纳米级别时，性能、能量效率和可靠性方面的主要瓶颈[7]。这主要因为线网的按比例缩小会增加电阻，并因而造成线网延迟和功耗的增加，同时更紧凑的间距影响信号完整性，从而造成可靠性问题。因此，设计可按比例缩小的高性能的、可靠的、能量效率高的片上互连是多核/SoC 设计范例成功的关键，

并已成为重要的致力研究方向。

传统上，基于总线的互连广泛应用于核数目较少的网络。然而，对于基于总线的互连，核数目的增长将成为其性能的设计瓶颈。于是，它们不被认为是未来具有多个核的多核系统的合理解决方案。为克服这些限制，一个可靠的选择是片上网络（NoC）[8,4]，其中通用片上互连网络通过使用开关矩阵或路由器来连接 IP 核或处理单元（PE），替代了传统的定制全局片上连线。典型的，处理器核使用分组交换协议互相通信，其将数据打包并通过片上网络传输。与传统宏网络类似，NoC 可规划性好。图 8-1 示出了 NoC 思想的一个概念图，其中许多核通过片上网络路由器连接，而不是通过片上总线。

尽管 3D 集成电路[32,15,29,30]和 NoC[8,4,23]都被认为是互连可规划需求的可替代方案，但将两种技术结合在一起设计 3D NoC 中的挑战直到最近才被意识到[14,10,5,34,33,16,17,21]。

第 7 章给出了使用 3D 集成（通过将缓存和功能单元分割到多层）的单核微处理器设计和使用 SRAM/DRAM 存储器堆叠的双核设计。然而，所有第 7 章中讨论的设计均不包含片上网络架构。在本章中，我们将关注如何结合 3D 集成和片上网络，以得到作为处理器核和存储器块之间的通信网络的结构。

在以下的章节中，首先给出 NoC 的简要介绍，随后讨论多种 3D 片上网络拓扑结构和 3D 路由器设计的方法，并且将给出一个使用 3D NoC 架构的存储器堆叠在多处理器芯片（CMP）上的实例。

8.2　片上网络的简要介绍

片上网络架构被认为是纳米时代出现的互连需求的一个可能的解决方案[8,4]。在片上网络架构中，通用片上互连网络通过使用开关矩阵或路由器连接 IP 核或处理单元（PE）代替了传统定制全局片上连线。PE 通过路由器发送信息包相互通信。这通常称为基于分组互连。

典型的 2D NoC 是由一定数量的处理单元（PE）排列在类似方格的网格结构（更像曼哈顿网格）中。PE 通过其下基于包的网络结构互连。每个 PE 都是通过网络接口控制器（NIC）连接到网络路由器。反过来，每个路由器连接到 4 个基本方向上的相邻的路由器。路由器端口的数目定义为路由器的阶数。

8.2.1　NoC 拓扑

网络拓扑是片上网络设计的一个重要方面，因为它决定了功耗—性能的度量标准。例如，NoC 拓扑决定了零负载延时、对分带宽、路由器微结构、布线复杂度、通道长度，以及整体网络功耗。

由于网格结构拓扑[4]（见图 8-1）的低复杂度和片面 2D 版图特性，其已经在

平面 CMP 很常见。这样一个简单固定的拓扑结构具有紧凑的 2D 版图。其他如集中网格和打平蝴蝶（flatten butterfly）的拓扑结构由于它们的多种优点也在 NoC 设计中有所应用。

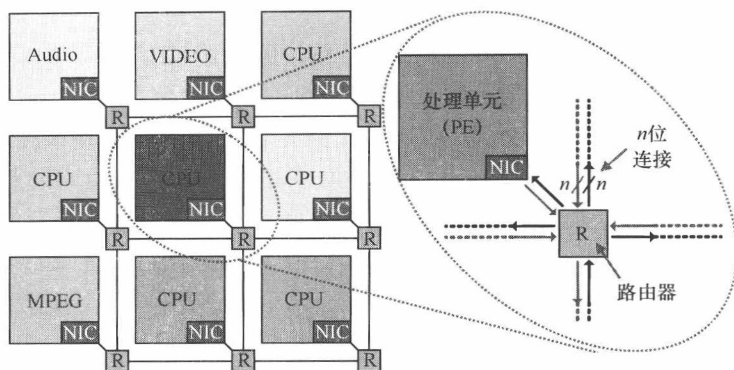


图 8-1　概念片上网络架构：核通过网络接口控制器（NIC）连接到片上网络路由器（R）

例如，集中网格（Cmesh）[2]保持了网格的优点，并通过共享多个处理元素间的路由器努力解决可规划性问题。共享路由器的节点数目称为网络的集中度。图 8-2 示出了 64 节点的 Cmesh 版图布局。这样的拓扑减小了路由器的数目，从而使跳跃数减小并得到网格上很好的延迟节省。Cmesh 阶数（端口数）为 8。由于其增长较慢的对分数，也可以提供非常宽的通道数（高于 5/2 位）。

另外一个例子是打平蝴蝶拓扑[9]，其通过同时采用集中方式和到非相邻节点的更长链接的方式来实现丰富的连接度，减小跳跃（hop 传递）数。更高的连接度增加了对分带宽并需要路由器上具有更多的端口（更高的阶数）。这里增加的对分带宽会导致通道数更窄。图 8-3 示出了一个 64 节点打平蝴蝶可能的版图布局。丰

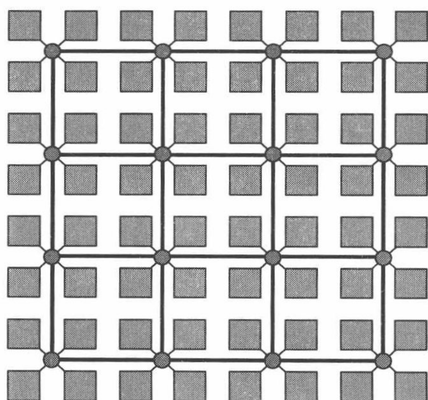


图 8-2　集中网格片上网络拓扑

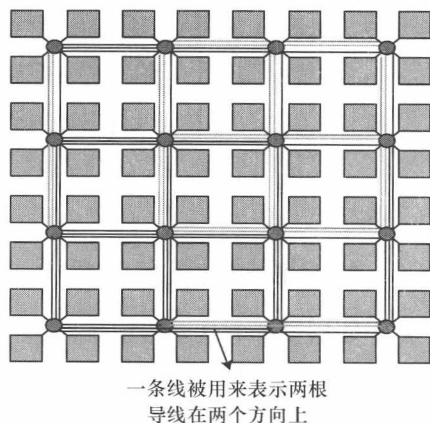


图 8-3　64 节点的打平蝴蝶拓扑

富的连接度通过减少跳跃数抵消了串联延迟。这样的拓扑阶数在 7 ~ 13 之间，取决于网络大小和较小的通道数（高于 128 位）。

8.2.2 NoC 路由设计

图 8-4 示出了一个基本的 NoC 路由器结构。路由器具有 P 个输入和 P 个输出通道/端口（端口的数目定义为阶数）。当 $P=5$（或阶数为 5）时，其为一个典型的用于网格结构的使用 5 × 5 交叉的 2D NoC 路由器。当网络拓扑改变时，路由器的复杂度也改变。例如，一个 CmeshC- 网格网络拓扑需要阶数为 8 的路由器设计。路由计算单元、RC、处理传入包的包头数据微片（数据微片是流控制的最小单元；包由多个数据微片组成），并基于包的目的地在选中的输出 PC 上给出合适的输出物理通道/端口（PC）和/或合法的虚拟通道（VC）。路由可以是确定性的或适应性的。虚拟通道分配单元（VA）在所有竞争访问到相同输出 VC 的包之间仲裁并选出胜利者。开关分配单元（SA）在所有要求访问交叉的 VC 间仲裁。胜利的数据微片可以随后穿过交叉并移动到它们对应的输出链。

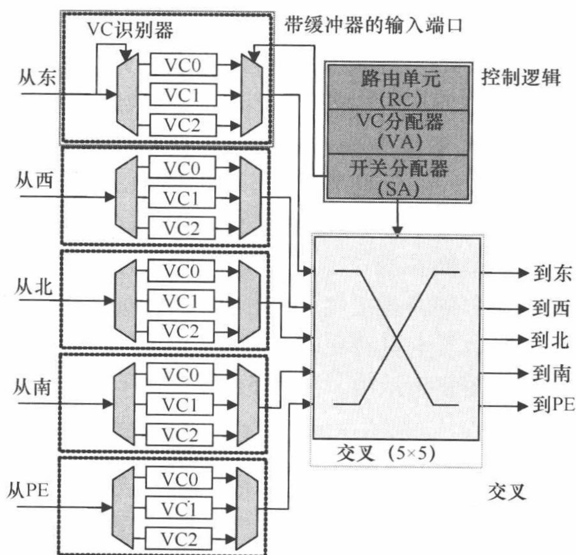


图 8-4 有 5 个输入端口和 5 个输出端口的基本 2D NoC 路由器

8.2.3 NoC 设计的更多信息

片上网络设计方法学吸引了众多工业关注。例如，Tilera Corporation 构建了 64 核嵌入式多核处理器 TILE64[1]，包含 64 个完整配置可编程核，使用基于网格的 NoC 架构连接。Intel 80 核 TeraFLOPS 处理器[31]组成了一个片上网络架构。该 80 核芯片以 8 × 10PE 核和包开关路由器阵列排列，使用网格拓扑（与图 8-1 类似）

连接。图8-5示出了处理器的NoC块结构框图。每个PE核包含两个流水线浮点乘法累加器（FPMAC），通过接口块（RIB）连接到路由器。路由器为带均步接口（MSINT）的基于交叉的5端口设计。网格NoC网络提供了2Tbit/s的对分带宽。

为了解更多的片上网络架构背景，可以参考书[8,4]和一些研究文献[19,23,12]。

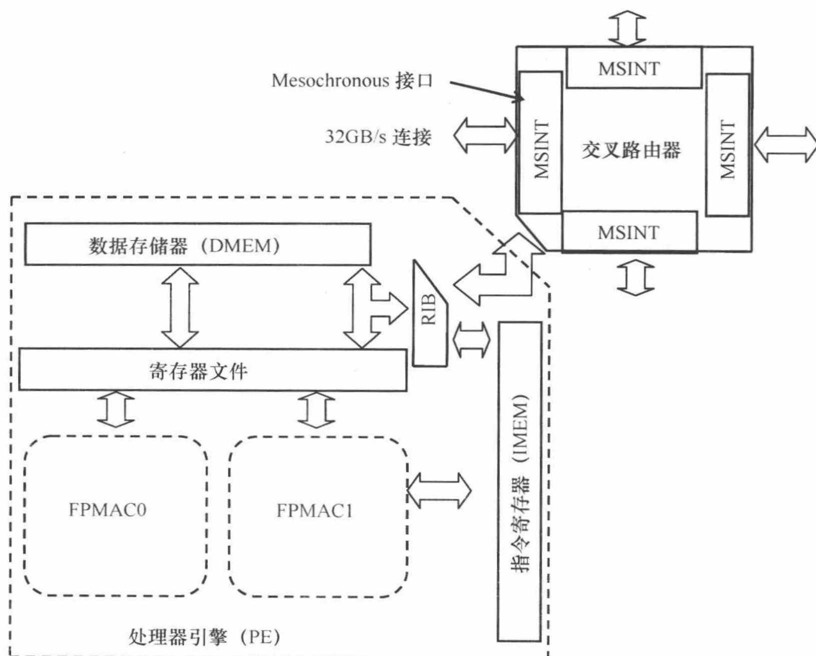


图8-5 Intel 80核TeraFLOPS处理器NoC框图

8.3 三维NoC架构

本节关注3D NoC架构可能的架构设计探索。将2D范例扩展到三维造成了有趣的设计挑战。已知片上网络受面积和功耗资源的严重约束，同时设计者还希望它们提供极低的延迟，那么关键的问题是如何在这些相对的设计线中取得合理的折中。在本节中，我们探索在考虑上述约束的情况下基本2D NoC如何实现向三维的扩展。

8.3.1 对称的NoC路由设计

让基本2D NoC路由器适应3D布局自然而简单的扩展就是简单地在每个路由器上加入两个附加的物理端口。一个负责“上”，一个负责“下”，再加上相关的缓冲器、仲裁器（VC仲裁和开关仲裁）以及交叉扩展。我们能通过在每层（因为所有方向的对称路由成为对称NoC）添加这样的路由器将一个传统NoC网络扩展

到三维。因为层内和层间移动具有相同的特征：逐次跳跃穿过（见图 8-6），我们称该结构为 3D 对称 NoC。例如，从 4 层芯片的底层移动到顶层需要 3 次网络跳跃。

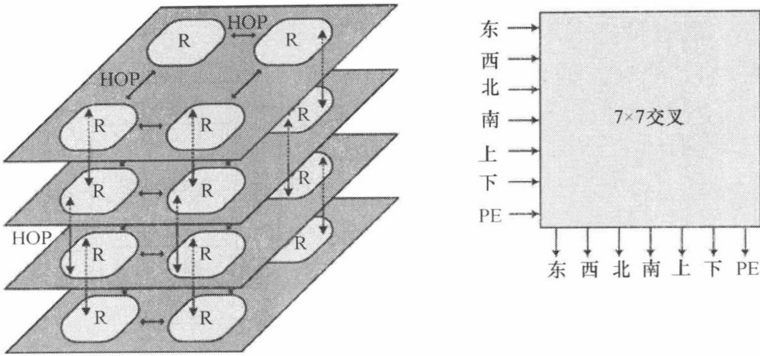


图 8-6 带有两个附加输入/输出端口（上方和下方）的对称 3D 片上网络路由器，总共需要 7 个输入端口和 7 个输出端口

这种结构虽然易于实现，但具有以下缺陷：

1）它浪费了 3D 芯片可忽略的层间距离这一优越特性（例如，在第 2 章中，我们发现晶片的厚度可以小到几十微米）。由于在垂直方向的穿越是多重跳跃，将消耗与层内移动相同的时间。当然，在源和目标间的跳跃平均数不会由于将 2D 设计折叠为多层堆叠就减少，但层间和层内跳跃是无法区别的。此外，在每次跳跃中每个数据微片必须通过缓冲和仲裁，这将增加层间上下移动的整体延迟。

2）添加两个附加的端口必须要一个更大的 7×7 交叉，如图 8-6b 所示。交叉变大效率很低，见表 8-1。表中包括本节中涉及的所有交叉类型在 90nm 工艺综合实现的面积和功耗的预算。显然，7×7 交叉造成了面积和功耗成本相对其他所有架构的显著增加。因此，从某种角度讲，3D 对称 NoC 实现是基本 2D NoC 网络幼稚的扩展。

表 8-1 在 90nm 工艺实现的交叉开关的面积和功耗比较

交叉类型	面积	功率/(500MHz)
5×5 交叉	8523μm^2	4.21mW
6×6 交叉	11579μm^2	5.06mW
7×7 交叉	17289μm^2	9.41mW

3）由于在 3D 架构中垂直和水平连接的不对称，有几个方面的问题，如连接带宽和缓冲分配，需要沿 3D 芯片的不同方向定制化。此外，在 3D 芯片不同层间的温度梯度和工艺变化会导致相同的路由元件在不同的层上出现不同的延迟。例

如，在距离热沉最远的层上工作元件将限制整个网络的最高频率。

8.3.2　三维（3D）NoC 总线混合路由设计

由于 3D 电路中快速的垂直互连和连接相邻核的水平互连间线网长度的差异（垂直方向为几十微米相对于水平方向的几千微米），3D 架构天生就具有非对称延迟。之前的章节证明，在垂直方向上（层间）具有多重跳跃通信的对称 NoC 架构是不需要的。

鉴于非常小的层间间距，单跳通信实际上是可行的。该技术反复考虑垂直距离相对于层内距离可忽略的事实；一个共享介质可提供任意层间的单跳穿越。该实现开启了到一个非常广泛共享的互连媒介——总线的大门。NoC 路由器可以在垂直方向与一个总线连接混合以创建如图 8-7 所示的 3D NoC 总线混合结构。该混合系统同时提供了性能和面积优势。替代了笨拙的 7×7 交叉的是 6×6 交叉（见图 8-7），即在基本的 2D 5×5 交叉的基础上增加一个单独的附加端口。附加的连接形成了 NoC 区域与总线（垂直）区域的接口。总线连接拥有其独立的专用队列。该队列由中央仲裁器控制。从不同层来的希望上/下移动的数据微片需要仲裁以获得共享介质的访问权。图 8-8 示出了垂直通孔结构的纵剖面图。该图说明了不同层间大的通孔压焊点没有用处；它们被故意做大来应对制造工艺过程中的对不准问题。结果是，巨大的通孔压焊点最终限制了 3D 芯片的通孔密度。

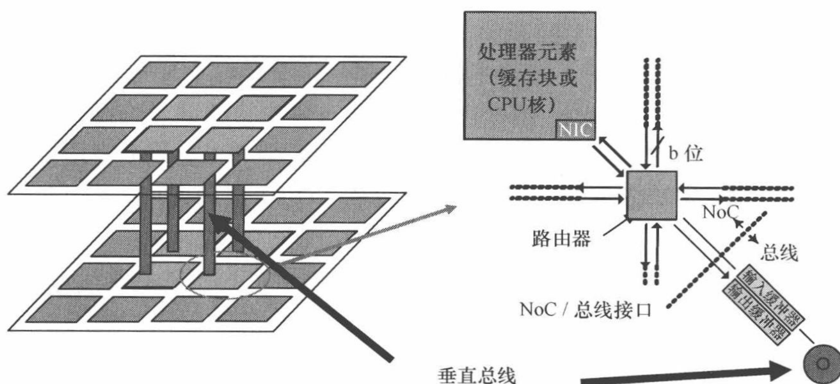


图 8-7　混合 3D NoC/总线架构。路由器具有一个附加输入/输出端口来连接垂直总线

除了 3D 说明的优于对称 NoC 路由器的优点，总线方式还受困于一个主要的缺陷，即它不允许在第三维并发通信。由于总线属于共享介质，在任意时刻它只能同时被一个数据微片使用。这严重增加了在高网络负载下竞争和阻塞的可能性。因此，虽然单跳垂直通信的确改善了整体延迟性能，但层间带宽受到损失。更详细的 3D NoC 总线混合结构的内容见本章参考文献［14］。

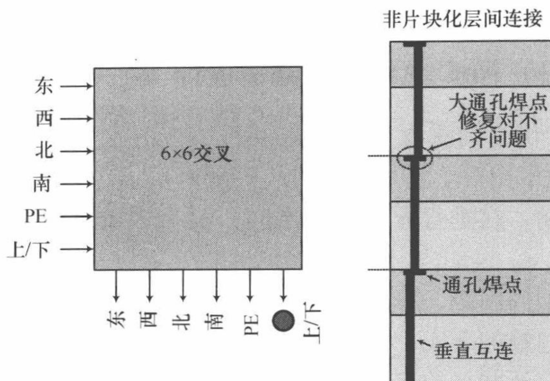


图 8-8 路由器具有一个附加输入/输出端口来连接垂直总线，并因此需要 6 个输入端口和 6 个输出端口。总线通过 3D 通孔实现多层互连

8.3.3 真三维（3D）路由设计

抛弃以前的选项，我们可以设想一个真正的 3D 交叉实现，其将实现垂直连接在整个路由器操作中的无缝集成。图 8-9 示出了这样一个 3D 交叉的版图布局。需要说明的一点是交叉的传统定义——在 2D 物理版图的背景下——是通过一个连接点实现每个输入连接到每个输出的开关。然而，将该定义扩展到物理 3D 结构就意味着一个十分复杂且尺寸巨大的开关（考虑到随多个层数而增加的输入输出端口对的数目）。因此，我们选择一个能提供通过超过一个连接点实现一个输入到一个输出端口互连的较简单的结构。虽然这样的配置可视为多级开关网络，但为简便起见，我们仍称其为交叉。垂直的连接现在嵌入在交叉中并扩展到所有层。这意味着 5×5 交叉的使用，不再需要附加的物理通道来专门用于层间通信。

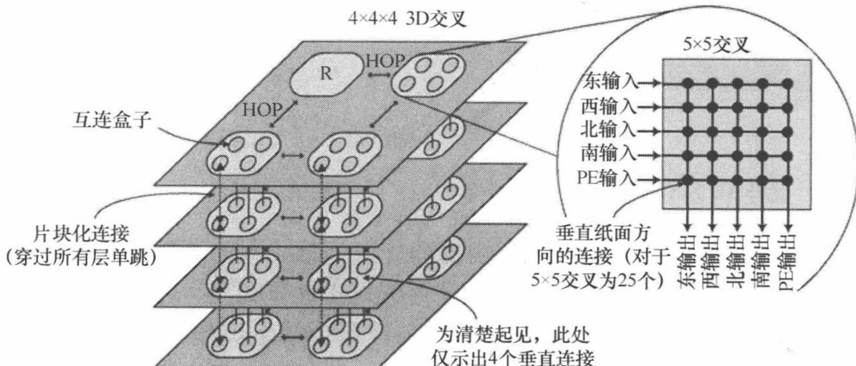


图 8-9 真 3D 路由器设计

从表 8-1 中可见，一个 5×5 交叉明显比 3D NoC 总线混合结构的 6×6 交叉和

3D 对称 NoC 的 7×7 交叉小并需要更少的功率。3D 交叉中的多种连接间的互连必须需要每层上有专用的连接盒子。这些连接点能使用垂直和水平通道的连接，允许 3D 交叉中灵活的数据微片遍历。这样一个连接盒子（CB）的内部配置如图 8-10 所示。图中垂直连接段也影响通孔版图布局。虽然该图远比图 8-8 所示的版图布局复杂，但偏离垂直通孔间的区域也能被其他电路使用，见图 8-10 中点线椭圆圈出的部分。因此，所有层的 2D 交叉物理上被融入一个单独的 3D 交叉。提供多重内部路径，一个数据微片穿过多个开关点通过并连接了输入和输出。此外，进入另外层的数据微片不需要穿过中间缓冲器；相反，它们直接连接到目标层的输出端口。例如，一个数据微片可以通过一个单跳从第 2 层的西侧端口输入，到第 4 层的北侧端口输出。

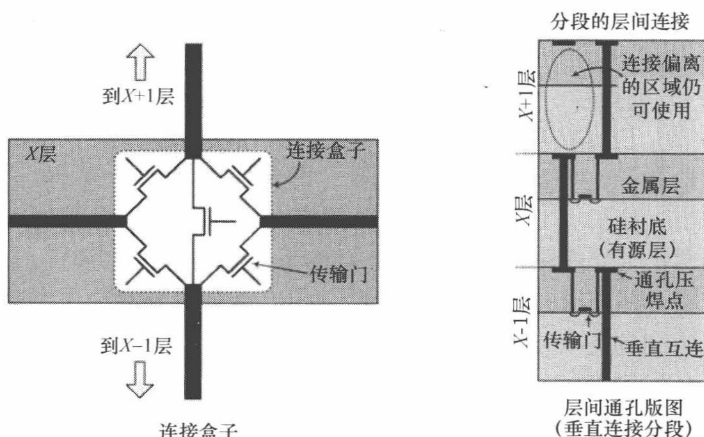


图 8-10　真 3D 路由器设计中 3D 交叉的层间通孔结构侧视图

然而，尽管这是个让人振奋的结果，而反面展现出一幅相当惨淡的画面。在 3D 交叉中增加大量的垂直连接来增加 NoC 连接度会导致路径多样性增加。这转化为来源和目标对间多个可能的路径。虽然这些增加的多样性开始看起来是好的特性，但它实际上将导致中央仲裁器复杂度的显著增加。这里，中央仲裁器即用来协调 3D 交叉的层间通信。仲裁器现在需要在多个可能的互连中做出决定并需要附加的大量控制信号来使能所有这些互连。即使仲裁器功能上可以分割为多个小的仲裁器，这些小的仲裁器间的协作将变得愈发复杂并占用更多时间。换言之，如果为了静态路径指定而牺牲动态性能，决定如何有效地指定哪些路径到每个源—目的对仍然很困难。此外，全 3D 交叉意味着每层 25（即 5×5）个连接盒子。一个 4 层的设计将需要 100CB！假定每个 CB 由 6 个晶体管组成，那么整个交叉结构仅传输门就将需要 600 个控制信号。这样的控制和连线复杂度将很大程度上注定主宰 NoC 路由器的整个操作。所有可能的输入输出组合的预编程静态控制序列将导致过大的表/索引；搜索这样一个表将引发巨大的延时，并同时造成面积和功耗的增加。大

量可能的连接反过来阻碍了开关流水线化的功能。注意到 NoC 路由器设计的主要目的是最小化操作复杂度来更好地实现非常短的流水线长度和极高的频率。一个具有压倒性的控制和协调复杂度的全交叉与朴素高效的设计方法学形成了鲜明的对比。此外，全互连提供的冗余度极少被现实中的工作负载使用到，也就是说，实际上此设计大材小用了[10]。

8.3.4 按维度分解 NoC 路由设计

给定 NoC 设计中紧凑的延迟和面积约束，垂直（层间）仲裁越简单越好。于是，一种真 3D 路由器设计，如以上章节所述，是不现实的选择。设计复杂度可以通过使用限制数量的层间链接得到缩减。本节将介绍一种模块化 3D 可拆解路由器（称为列—行—垂直路由器（RoCoVe））[10]。

在典型的 2D NoC 路由器中，5×5 交叉有 5 个输入/输出，对应着平面的 4 个方向和到本地 PE 的连接。交叉是路由器延迟和面积增加的主要原因。本章参考文献［11］中示出，通过已知的如引导数据微片队列的基本开关处理的使用，输入流量可分解为两种独立的流：①东西流量（即包在 X 方向的移动）；②南北流量（即包在 Y 方向的移动）。这样的流量隔离策略允许使用更小的交叉并且此策略隔离的两种流到两个独立的路由器子模块，被称为行模块和列模块[11]。

采用与流量分解相同的思路，3D NoC 中的流量可以分解为 3 个独立的流，即加入第三个 Z 方向的流（层间通信）。在三维中处理所有的流量需要一个附加的模块，此模块称为垂直模块。此外，必须要有在垂直模块和行/列模块间的连接来保证包从垂直模块到行/列模块的移动。于是，这样一种方向分解的方法允许使用较小的 4×2 交叉，从而实现了更快和更高效的 3D NoC 路由器设计。3D 方向分解 NoC 路由器设计的结构图如图 8-11 所示。更多详细的内容参见本章参考文献［10］。

8.3.5 多层三维 NoC 路由设计

之前介绍的所有 3D 路由器的设计选择（对称 3D 路由器、3D NoC 总线混合路由器、真 3D 路由器和 3D 方向分解路由器）均基于处理单元（PE）（为处理器核或缓存块）本身仍是 2D 设计的基础上的。在 7.4 节中，介绍了一种微处理器的细粒度设计，可以将 PE 分成多层实现。例如，3D 缓存设计和 3D 功能单元设计。第 4 章也讨论了三维块设计和布局规划算法。于是，一个 NoC 结构中的 PE 可以采用这种细粒度方法实现。虽然这样的 PE 多层堆叠在目前的工艺下实现较勉强，但当 3D 工艺成熟且 TSV 间距更小时将成为可能（如 7.4 节中介绍的）。

采用这样的 NoC 架构下的多层堆叠 PE，设计充斥 3D 芯片所有层的多层 3D 路由器成为必须。即使每个 PE 和路由器的面积更小，路由器间的距离更短，但从逻辑上看，这样的含多层 PE 的 NoC 架构和多层路由器与具有同样多节点的 2D NoC 情况类似。于是，与 2D 路由器相比，多层路由器的设计不需要附加功能，仅需要

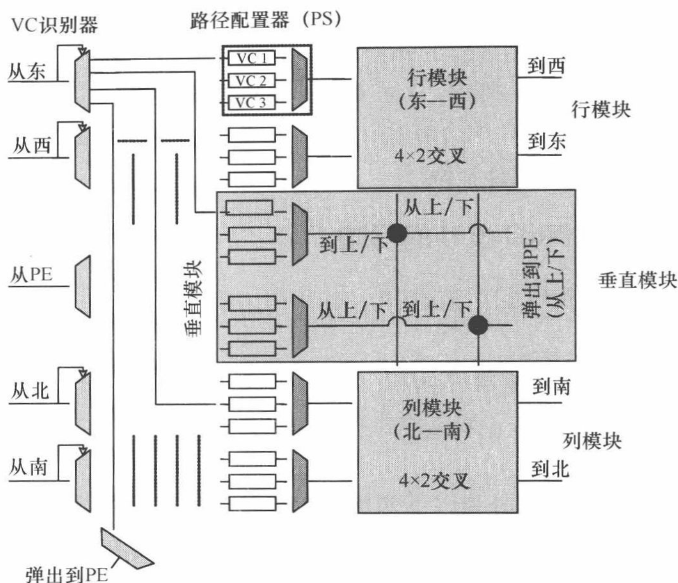


图 8-11　3D 合成 NoC 路由器设计详细架构

使得功能分布在多层上。

基于可系统地将模块划分为跨层的带有层间连线约束更小的子模块的能力和平衡层间面积的需要，路由器模块可分为两类：可分割的和不合分割的[5]。输出缓冲器、交叉和路由器间连接属于可分割的模块，而仲裁逻辑和路由逻辑由于不能系统地分割为子集，属于不可分割模块。节省的芯片面积可用于提升路由器容量，比如，增加非邻接 PE 间快速路径来减少平均跳跃数，同时有助于提升性能并降低功耗。此外，因为大部分的通信流量由短数据微片和常见向量组成，所以可以通过动态关闭多层路由器的某层来降低功耗。

8.3.6　三维 NoC 拓扑设计

到现在为止，讨论的所有路由器设计均基于以网格为基础的 NoC 拓扑结构。如 8.2 节所示，当今有多种拓扑结构，像中心网格和打平蝴蝶拓扑，它们分别有自己的优缺点。通过采用不同的拓扑结构而不是网格拓扑，以上讨论的路由器设计会出现不同的变化。例如，在 2D 中心网格拓扑中，路由器本身阶数为 8（即 8 端口路由器，4 个连接本地 PE，其他地连接到 4 个水平方向）。在这样的拓扑结构下，3D NoC 总线混合方法将生成 9 端口路由器设计。这样的高阶路由器虽然减少了在 PE 间的跳跃数，但整个设计的功耗大、性能低。于是，3DNoC 需要一种拓扑—路由器协同设计方法以使任意两个 PE 间的跳跃数和 3D 路由器设计的阶数尽可能小。Xu 等在本章参考文献［33］中提出了一种带有小尺寸低阶路由器设计的 3D NoC

拓扑结构。水平 2D 网格被连接至少在 m 个网格跳跃之外的节点的长连接网络所替代，其中 m 是设计参数。在这样的拓扑中，长距离通信可以利用长物理线网和垂直连接实现。对于专用 NoC 架构，Yan 和 lin[34] 也提出了称为向上跳重布线和路由器融合（RRRM）的 3D NoC 综合算法，该算法基于布线流程的布线重排规则和为减少跳跃数而设计的路由融合步骤。

8.3.7　三维工艺对 NoC 设计的影响

第 2 章讨论了 3D 集成的工艺选择。本节中，将重点讨论多种 3D 集成方法对 NoC 设计的影响。

由于 TSV 通孔竞争有源器件面积，故必须约束单位面积内这样通孔的数目。于是，NoC 设计必须与其他（如电源供电网络和时钟网络等）竞争同样互连资源的系统组件设计协同进行。

使用 TSV 的 3D 集成可分为以下两类：①单片方法；②堆叠方法。第一种方法包含一系列器件工艺，其中前端工艺（构建器件层）反复在一个晶圆上进行以在后端工艺构建器件间的互连之前构建多个有源器件层。第二种方法（可能为晶圆—晶圆、晶片—晶圆或晶片—晶片堆叠）使用传统制造技术分别处理每个有源器件层。这些多重器件层随后使用绑定技术封装以构建 3D IC。晶片可以面对面（F2F）或面对背（F2B）绑定。微焊点包在面对面晶圆绑定中不穿过厚的掩埋硅层并可以采用高间距密度制造。对与堆叠绑定，由于对准容差和在绑定过程中减薄的晶片/晶圆高度限制了 TSV 的按比例缩小，故 TSV 的尺寸不能与特征尺寸同步按比例缩小。

TSV（或微压焊点）的大小、长度和间距密度和绑定方法（面对面或面对背绑定，基于 SOI 的 3D 或基于体 CMOS 的 3D）对 3D NoC 拓扑设计有巨大的影响。例如，TSV 相对较大的尺寸会阻碍在多个器件层间以较细的粒度分块的设计过程，并使真 3D 路由器设计的可能性降低。另一方面，单层 3D 集成中，垂直 3D 通孔由于采用本地线网互连而有可能与特征尺寸同步按比例缩小，这为设计提供了更灵活的垂直 3D 互连。这样工艺的存在使以极细的粒度分块成为可能。此外，相比面对背的绑定或基于体 CMOS 的集成，面对面绑定或基于 SOI 的 3D 集成可能具有更小的通孔间距尺寸和更高的通孔密度。这些 3D 工艺参数对 NoC 拓扑的影响将被进一步研究，并且关于性能、功耗、热和可靠性优化适用于不同 3D 工艺的 NoC 拓扑将被确定。

8.4　使用三维 NoC 架构的多处理器芯片设计

在以上章节，讨论了用于 3D NoC 架构的多种路由器设计和拓扑探索。本节中，我们以 3D NoC 总线混合架构为例研究采用 3D NoC 架构的带存储器堆叠的芯

片多处理器设计，并同时评估该架构设计的益处[14]。

多核在单芯片上的集成被认为会加重本已糟糕的存储器带宽问题。为含大量片上核的芯片提供足够的数据将成为性能可规划度的主要挑战。受I/O引脚所限，传统的片上存储器将不足。根据ITRS预测，在下个十年中，封装上引脚的数目将不能持续增长以满足克服该问题的需要。于是，预计在多核之上堆叠存储器将成为3D工艺的早期商业应用。

在不远的将来，在采用CMP和其他多核系统时有望增加二级和三级缓存的尺寸。然而，由于微小的特征尺寸会恶化互连延迟的影响，而成为达到设计的性能和功耗预算的关键瓶颈。因此，虽然传统架构在存储器层次的每层都具有单个、一致的访问时间，但互连延迟的增加将根据所要求的缓存线的物理位置而改变大缓存的访问时间。也就是说，访问时间将转换为基于芯片上穿越距离的可变延迟。

基于以上的观察，本章参考文献［3］中提出了非一致缓存架构（NUCA）。与大的标准单片二级缓存不同，NUCA中的L2空间分为多个块，并根据其与处理器的相对位置分别具有不同的访问延迟。这些块通过基于网格的互连网络连接。缓存线允许在这些网络中迁移以将更常访问的数据放置在离处理器较近的块中。几个最近提出的方案将NUCA的理念扩展到CMP中。在CMP架构中NUCA先天的问题是被多个核共享的数据管理问题。已经提出的解决该问题的方案包括数据复制和数据迁移。同时，长的访问延迟和高功耗也仍然是基于NUCA的CMP的固有问题。

3D电路的引入提供了减小线网长度并增加存储器带宽的可能。所以，该技术可以有效减少NUCA架构中远离处理器核的缓存块的访问延迟。

7.2节中讨论了在不需要使用NoC架构或NUCA架构的理念情况下双核处理器中堆叠SRAM和DRAM二级缓存的设计。本节中，我们考虑NUCA的3D设计，其结合了NoC和3D技术的优点以减小基于CMP的系统中二级缓存的访问延迟。本节提供了3D NoC拓扑设计的新视角，强调了二级缓存中数据管理的问题，并同时考虑了网络流量和热的问题。

8.4.1 三维二级缓存在CMP架构上的堆叠

在第1章中提到了，3D芯片的一大优点是层间较小的距离。在第2章中，我们看到两层间的距离处在几十微米的量级，即相对于2D片上网络架构中两个NoC路由器间穿行的距离是可忽略的（例如，65nm工艺下实现的64KB混存块中平均为1500μm）。该特性使垂直（层间）方向的通信比水平（层内）方向快得多。本节中，我们将介绍在CMP处理器上堆叠二级缓存的架构，在这种结构中，3D技术使从CMP处理器到堆叠的二级缓存的快速访问成为可能。

如8.3节中讨论的，直接的3D NoC路由器设计是对称3D NoC路由器，其增加了设计复杂度（使用7×7交叉），并导致了在非邻接层间多次跳跃通信。NoC总线混合设计不仅减小了设计复杂度（使用6×6交叉），而且也因为层间较短的

距离实现了在层间的单跳通信。本节将重点介绍 NoC 总线混合架构，其使用动态时分多址（dTDMA）总线作为晶圆间的“通信柱”，见图 8-7。这些垂直的总线柱使用最少的硬件在任意两层间提供单跳通信并转接到传统 NoC 路由器实现层内传递，这将在下文详细讨论。由于工艺的限制和路由器复杂度问题（随后讨论），不是所有的 NoC 路由器均可包含垂直总线，但含垂直总线的就形成了到其他层的网关。因此，那些连接到垂直总线路由器的架构稍有修改。

8.4.2 dTDMA 总线作为通信支柱

dTDMA 总线架构[24]消除了通常与总线相伴的进程特征，相反，采用了动态增大或缩小时间沟槽数目的仲裁器以匹配活动客户数目。单跳通信和进程较少的仲裁允许低的可预测的延迟。动态分配一直产生最高效的时间沟槽配置，使 dTDMA 总线带宽效率达到将近 100%。每个柱节点需要紧凑的收发器模块来转接总线，如图 8-12 所示。

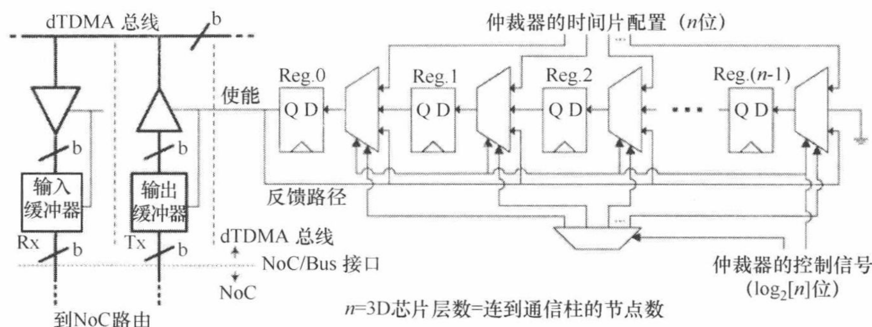


图 8-12 dTDMA 总线的收发模块

dTDMA 总线接口（见图 8-12）包括通过三态驱动器连接到总线的发送器和接收器。在每个接收器和发送器上的三态驱动器由独立编程全抽头反馈移位寄存器控制。因为其较小的尺寸，dTDMA 总线接口是 NoC 路由器最小的附加部分。

中心化仲裁器的出现是芯片中垂直总线或柱需要保持较低数目的另一个原因。每个带有连接所有层的控制信号均需要一个仲裁器。仲裁器应置于芯片的中间层来保持连线距离尽可能的一致。自然地，控制线的数目随挂在柱上的柱节点数（即芯片中的层数）的增加而增加。dTDMA 总线架构仲裁器和所有其他的单元均用 Verilog HDL 实现，并用商用 90nm TSMC 库综合。仲裁器和收发器占用的面积远小于 NoC 路由器，因此彻底证明了使用该方法作为层间垂直网关的正确性。这些 dT-DMA 组件和基本 5 端口（东、西、南、北、本地）NoC 路由器的面积和功耗（均在 90nm 工艺下综合）见表 8-2。显然，添加 dTDMA 组件的面积和功耗的代价成量级的小于整体预算。因此，使用 dTDMA 总线作为垂直互连有最小的面积和功耗影响。dTDMA 总线被视为在层数小于 9 时垂直方向优于对称 3D 路由器设计的方案。

如在第2章中讨论的，TSV的寄生参数由于其较小的尺寸对功耗和延迟的影响较小。层间通孔的密度决定了可采用的柱的数目。表8-3示出了由170个线网（128位总线+4层3D SoC所需的3×14控制线网）构成的柱在不同的通孔间距尺寸下所占用的面积。在面对背的3D实现中，柱必须通过有源器件层，这意味着柱占用的面积即为浪费的器件面积。这就是层间互连数目必须最小化的原因。然而，随着通孔密度的增加，柱占用的面积越来越小，并与NoC路由器所占用的面积相比逐渐可忽略（见表8-2和表8-3）。然而，如第2章中提到的，通孔密度仍将受到通孔压焊点尺寸的限制，不能随实际通孔尺寸同步按比例缩小。如表8-3所示，即使间距为5μm，对于基本5端口NoC路由器，柱引入的面积成本约为4%，并不显著。这些结果说明，出于我们3D架构的目的增加额外的dTDMA总线柱是可行的。

表8-2 dTDMA总线的面积和功耗代价

元件	功率	Area/mm^2
基本NoC路由器（5端口）	119.55mW	0.3748
dTDMA总线Rx/Tx（每用户2个）	97.39μW	0.00036207
dTDMA总线仲裁器（每总线1个）	204.98μW	0.00065480

表8-3 不同通孔间距尺寸下晶圆间连线的面积代价

总线宽度	晶圆间面积（根据dTDMA总线连线）			
	10μm	5μm	1μm	0.2μm
128bits（+42control）	62500μm^2	15625μm^2	625μm^2	25μm^2

然而，通孔密度不是限制柱数目的唯一参数。路由器的复杂度也扮演了关键角色。如上所述，在NoC路由器中添加额外的垂直链接（dTDMA总线）将使端口数从5增加到6，并且由于在每个路由器中的竞争概率直接与竞争的端口数成正比，端口的增加将增加竞争概率。这反过来将增加路由器中的拥塞，原因是将有更多的数据微片到路由器交叉的访问需要仲裁。因此，任意地在NoC路由器中添加垂直柱反过来影响了每个柱路由器的性能。于是，网络中高竞争路由器数目（柱路由器）增加，其附近层内和层间通信的延迟跟着增加。

另一方面，存在最小可接受的柱数目。我们在该研究中将每个CPU置于其自己的柱之上。如果多个CPU可以分享同一个柱，就会存在更少的柱，但这样的组织将引发如竞争等其他问题。

8.4.3 三维（3D）NoC总线混合路由架构

8.2节提供了3D NoC总线混合路由器设计的简要介绍，本节将给出详细的设计。

一个基本的NoC路由器由4个主要部分组成：路由单元（RT）、虚拟通道分配

单元（VA）、开关分配单元（SA）以及交叉（XBAR）。在网格拓扑中，每个路由器有 5 个物理通道（PC）：东、西、南、北和连接本地处理单元（CPU 或缓存块）通道。每个物理单元具有一个虚拟通道（VC）数。这些是保存发来不同消息数据微片的先进先出（FIFO）缓冲器。在实现中，我们采用了每个 PC 含 3 个 VC，每个 VC 含 1 消息，每个消息为 4 个微片长。每个消息选定为 4 数据微片长。路由器链接的宽度选定为 128 位。结果，一个 64B 缓存线可放入一个包（即 4 数据微片/包 ×128 位/数据微片 =512bit/包 =64B/包）。

最基本的路由器实现是 4 级的，即在路由器中的每个单元需要 1 个时钟周期。在我们的二级架构中，低网络延迟是最重要的，也就是更快的路由器是必需的。本章参考文献［22］提出了更低延迟的路由器架构，此架构采用称为预测分配的技术使 RT、VA 和 SA 并行。该方法在 VA 阶段预测胜出者并基于此执行 SA。此外，称为预查找路由的技术也用于执行提前一步的路由（在节点 i 执行节点 $i+1$ 的路由）。这两种修改可以有效地改善路由器的性能。通过并行多阶段的操作，两阶段甚至单阶段[20]的路由器成为可能。在我们提出的架构中使用了单阶段路由器以最小化延迟。

由于在 dTDMA 柱和 NoC 路由器间必须有接口以实现在层中 2D 网络垂直连接的无缝集成，连接到柱节点的路由器是不同的。修改的路由器见图 8-7。一个额外的物理通道 PC 加入路由器，其用于垂直连接。额外的 PC 有其自己的专用缓冲器并与其他到路由器操作的连接没有差别。路由器仅看到一个附加的物理通道。

8.4.4 处理器和二级缓存组织

图 8-13 示出了在我们设计中的处理器和二级缓存的组织结构。与 CMP-DNUCA[3]类似，我们将缓存块分为多个簇。每个簇包含一套缓存块和对应簇中每个缓存线的独立的标签矩阵。有些簇中心放置了处理器，有些则没有。簇中所有的块通过 NoC 连接来实现数据通信，而标签矩阵直接连接到簇中的本地处理器。注意到每个处理器有其自己专用的一级缓存和一个在其本地簇中的带标签矩阵的二级缓存块。对于不含处理器的簇，标签矩阵连接到用于接收缓存线要求，查找标签矩阵和将要求传递到目标缓存块的定制逻辑块。该处理器和缓存的组织结构可通过改变尺寸和/或簇数，按比例缩小。

8.4.5 缓存管理策略

基于上一节给出的处理器和缓存的组织结构，我们开发了缓存管理策略，包括缓存线查找策略、缓存放置和替换策略，以及缓存线迁移策略，所有将在以下章节中详细介绍。

8.4.5.1 查找策略

缓存线查找策略是两步过程。在第一步中，处理器查找其所属簇中的本地标签

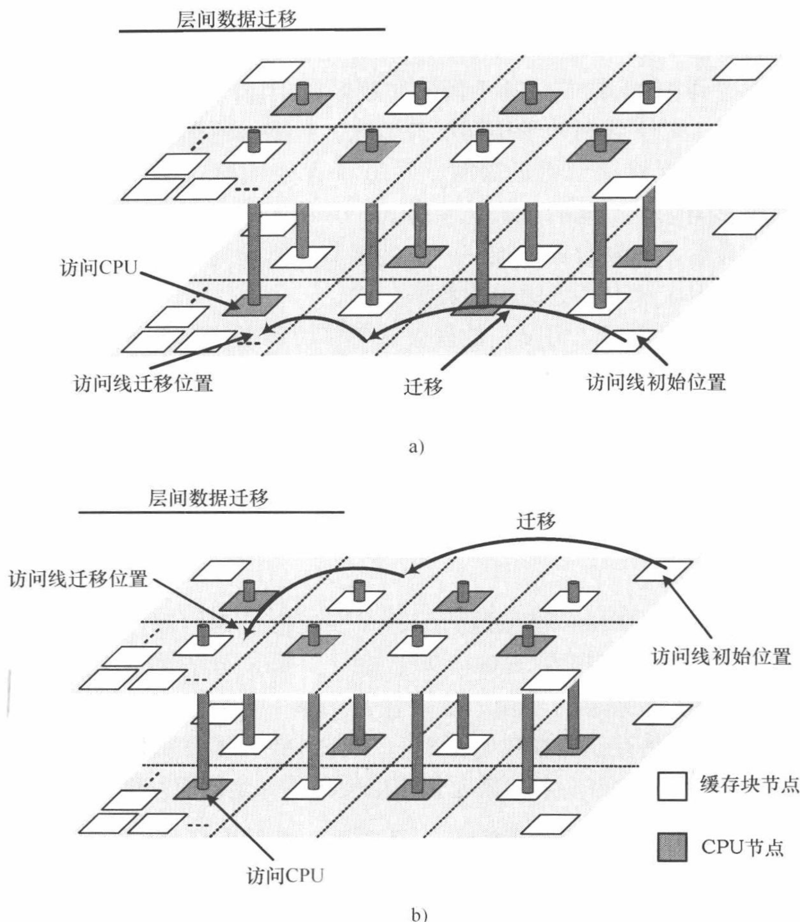


图 8-13　在 3D 二级架构中层内和层间的数据迁移。点连线代表簇

矩阵并发出查找其相邻簇中标签矩阵的要求。所有垂直相邻的簇，通过柱广播的标签接收。如果缓存线在这些位置均找不到，那么处理器将要求广播到剩余的簇。如果标签在所有簇中均不匹配，那么视为二级缓存丢失。在任一簇中匹配的标签之上，对应的数据通过 NoC 路由至发出请求的处理器。

8.4.5.2　放置和替换策略

我们使用与 CMP-DNUCA[3] 相似的放置和替换策略。初始阶段，缓存线根据其缓存标签低位放置；即这些位决定缓存线初始将被放置的簇。缓存索引的低位指向簇中缓存线将被放置的块。缓存索引剩余的位决定了其在缓存块中的位置。簇的标签标志也在缓冲线放置时刷新。放置策略仅用于决定缓存线的初始位置，因为当缓存线开始迁移时，缓存标签的低位不再指向簇位置。最终，我们使用伪 LRU 替换策略来驱除缓存线以实现缓存丢失。

8.4.5.3 缓存线转移策略

与之前的方法类似，我们的策略试图将数据迁移得离访问处理器更近。因此，我们的策略裁剪到3D架构并且基于访问的数据处于访问处理器的同层或不同层来区别地处理迁移。对于数据处于相同层的情况，数据被逐渐迁移到距访问处理器更近的簇。当将缓冲线移动到更近簇时，跳过了含处理器（非访问处理器）的簇，因为我们不想影响它们本地的二级访问样式，并将缓冲线移动到下一个不含处理器的最近的簇。最后，如果数据被唯一一个处理器反复访问，它将移动到该处理器的本地簇中。图8-13a示出了该层内的数据迁移。

对于数据处于不同层的情况，数据被逐渐移动到更接近距离访问处理器最近的柱的位置（见图8-13b）。因为可通过垂直通信柱访问的簇被认为处于本地近处，不会跨层移动数据，这样有利于减小缓存线移动频率，反过来也节省了功耗。

为了避免错误丢失（因在迁移过程中查找数据造成的丢失），我们采用了CMP-DNUCA[3]中使用的懒惰迁移机制。

8.4.6 方法学

我们使用连接了3D NoC模拟器的Simics[18]模拟了3D CMP架构。一个8处理器CMP架构的全系统模拟在Solaris 9上执行。每个处理器使用顺序流并执行SPARC ISA指令集。处理器含专用一级缓存并共享一个大的二级缓存。默认配置的处理器、存储器和网络存储器参数见表8-4。该表中的一些参数为研究不同的配置进行了修改。示出的缓存块和标签矩阵延迟提取使用了知名的缓存模拟器Cacti[27]。

表8-4 默认系统配置参数（二级缓存按16簇16×64KB组织）

处理器参数	
处理器数目	8
发射宽度	1
存储器参数	
一级（I/D分开）	64KB，2路，64位线，3周期写出
二级（统一）	16MB（256×64KB），16路，64位线，5周期块访问
标签阵列（每簇）	24KB，4周期访问
存储器	4GB，260周期延迟
网络参数	
层数目	2
柱数目	8
布线方案	尺寸顺序
开关方案	虫洞
飞掠区大小	128位
路由器延迟	1周期

为建模3D、混合NoC/总线互连的延迟，我们在现有2D NoC模拟器[12]的基础上开发了基于C的精确周期模拟器。为了该研究，2D模拟器扩展到了三维，同时dTDMA总线集成作为垂直通信通道。3D NoC模拟器作为输出，生成了缓存访问延迟。

在我们的缓存模型中，不同处理器的专用一级缓存通过实现分布式的基于目录协议保持一致。每个处理器有跟踪其一级缓存线状态的目录。基于MESI协议，一级访问事件（如读丢失）引发状态转换并更新到目录。由一级缓存一致引起的流量在我们模拟的考虑之中。

使用模拟平台，我们模拟了9个SPEC OMP基准电路[28]。对于每个基准电路，我们在源代码中标记了初始相位。在初始化完成后缓存模型才开始模拟。之后，每个应用执行5亿周期来预热二级缓存。随后我们收集接下来20亿周期的统计数据。

8.4.7 结果

我们首先介绍了在实验中比较的方法。参考了CMP-DNUCA[3]中的完美查询方法。我们将2D和3D方法分别命名为CMP-DNUCA-2D和CMP-DNUCA-3D。注意2D方法仅是本章中讨论的3D方法的特例，即仅含一层。这两种方法都采用了缓存线迁移。为屏蔽3D工艺的有益影响，我们还实现了不含缓存线迁移的3D方法，称为CMP-SNUCA-3D。

第一套结果给出了在不同方法下平均二级选中延迟数。结果如图8-14所示。结果表明（CMP-DNUCA-2D）生成了与之前2D方法（CMP-DNUCA[3]）相近的结果。因为处理器没有像CMP-DNUCA那样放置在芯片的边缘，而是如图8-13所示的用缓存包围着它们，所以2D方法在几个基准电路上显示出略好的IPC结果。3D方法的结果显示出了预期的本地性的增加所带来的优势。有趣的是CMP-SNUCA-3D没有采用迁移，却仍胜过采用迁移的2D方法。平均来看，当从CMP-DNUCA-2D变到CMP-DNUCA-3D时，二级缓存延迟缩减了10个周期。在3D工艺中使用数据迁移得到更好的性能也是可能的。特别地，CMP-DNUCA-3D比静态3D平均减少了7个周期的延迟。此外，我们注意到即使当采用迁移时（见图8-15），由于增加的本地性，3D使用迁移的次数少于2D。缩减的迁移数反过来减小了网络流量和功耗。为了与二级访问的次数相称，这些二级延迟节省翻译到IPC改善上。图8-16示出了CMP-DNUCA-3D（CMP-SNUCA-3D）优于2D方法，其中IPC改善达到37.1%（18.0%）。IPC改善高于mgrid、swim和wupwise，由于这些应用展现了较高的二级访问数。

我们接着研究了较大缓存尺寸对使用CMP-DNUCA-2D和CMP-DNUCA-3D所带来节省的影响。当增加二级缓存尺寸时，为了保持16路协同性，增加了每个簇的尺寸。图8-17示出了32MB和64MB二级缓存在4个有代表性的基准电路上的平

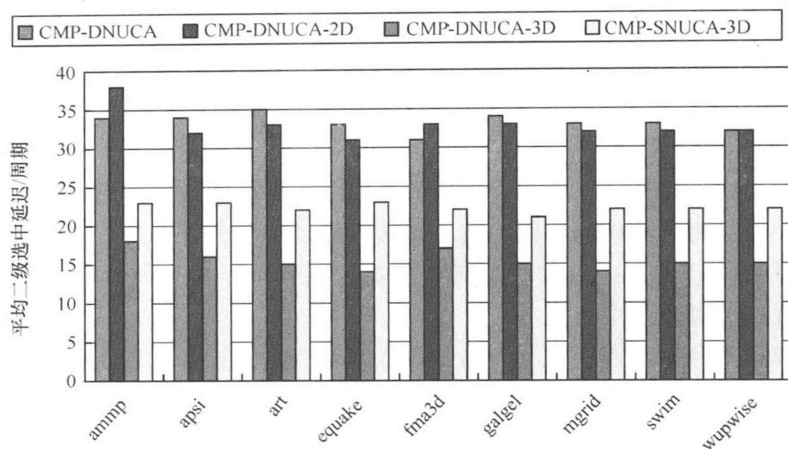


图8-14　不同方案下平均二级选中延迟值

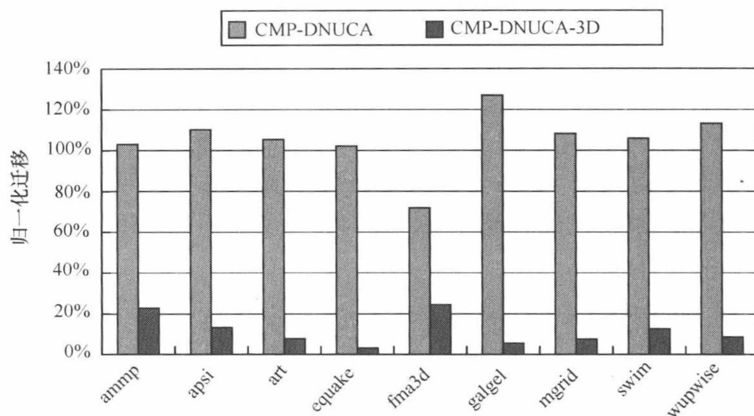


图8-15　CMP-DNUCA和CMP-DNUCA-3D的块迁移数目，根据CMP-DNUCA-2D归一化

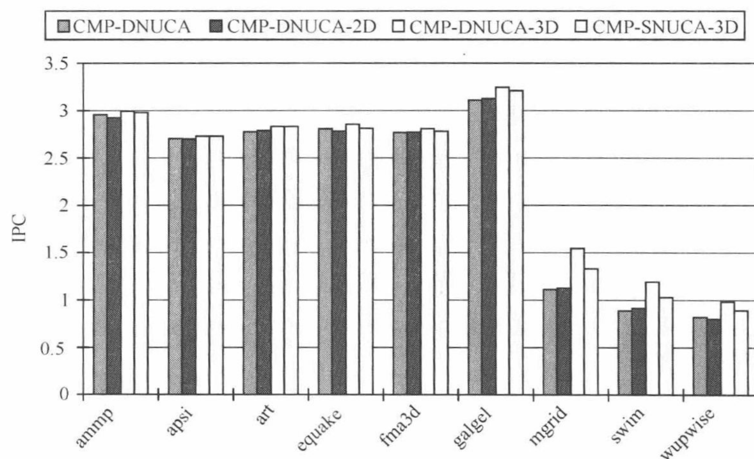


图8-16　不同方案下IPC值

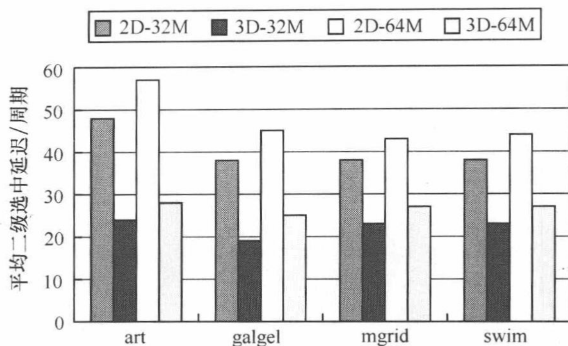


图 8-17　不同方案下平均二级选中延迟值

均二级延迟结果（art 和 galgel 一级未命中率低，mgrid 和 swim 一级未命中率高）。我们观察到二级延迟随大缓存尺寸增加，即使在 3D 配置下的较慢速率（2D 平均 7 周期，3D 平均 5 周期），也显示 3D 拓扑是采用更大二级尺寸时更可按比例规划选择。

之后我们通过在已知 3D 拓扑下修改一些参数展开实验。用 CMP-DNUCA-3D 方法，使用不同柱数以捕捉不同层间通孔间距对其影响的结果如图 8-18 所示。随着柱数的减少，共享资源（柱）的竞争会增加，以保证层间通信。结果是，当从 8 柱改到 2 柱时，平均二级延迟从 1 周期增加到 7 周期。当层数从 2 增加到 4 时，二级延迟减小了 3 ~ 8 个周期，主要由于访问数据的距离缩短，对其的说明如图 8-19 中的 CMP-DNUCA-3D 方法。

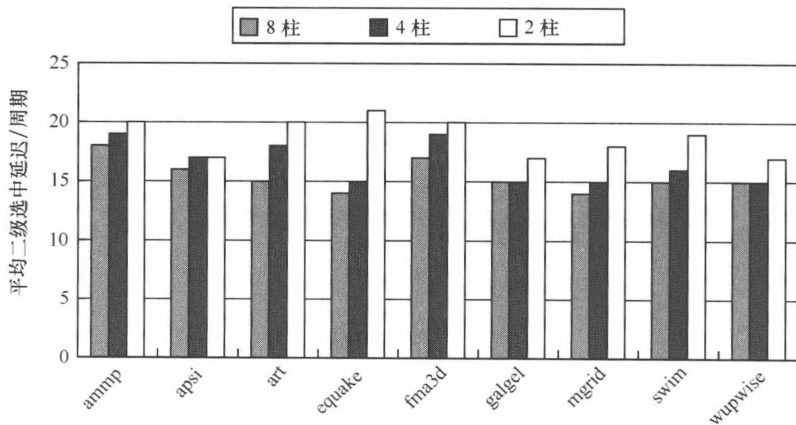


图 8-18　柱数目的影响（CMP-DNUCA-3D 方案）

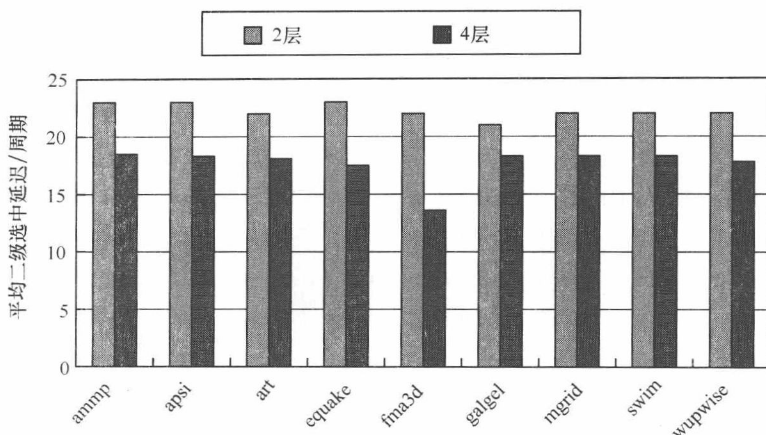


图 8-19 层数的影响（CMP-SNUCA-3D 方案）

8.5 结论

三维电路和 NoC 是互连复杂度增长下的两个前沿趋势。在本章中，我们介绍了设计 3D NoC 架构的多种方法并展示将 NoC 和 3D 架构结合是设计未来多处理器芯片的一大可靠选择。

致谢

这里展示的大量工作和想法是和我的同事及研究生多年研究的成果，其中特别感谢 Mahmut Kandemir 教授，Intel 公司的 Mazin Yousif 博士，Chrysostomos Nicopoulos，Thomas Richardson，Feihui Li，Jongman Kim，Dongkook Park，Reetuparna Das，Asit Mishra，Soumya Eachempati。本研究受 NSF 基金，EIA-0202007，CCF-0429631，CNS-0509251，CCF-0702617，CAREER 0093085，以及 DARPA/MARCO GSRC 的大力支持。

参考文献

1. A. Agarwal, L. Bao, J. Brown, B. Edwards, M. Mattina, C. Miao, C. Ramey, and D. Wentzlaff. Tile processor: Embedded multicore for networking and multimedia. In *Proceedings of Hot Chips Symposium*, 2007.
2. J. Balfour and W. J. Dally. Design tradeoffs for tiled CMP on-chip networks. In *Proceedings of International conference on Supercomputing*, pp. 187–198, 2006.
3. B. M. Beckmann and D. A. Wood. Managing wire delay in large chip-multiprocessor caches. In *Proceedings of International Symposium on Microarchitecture*, pp. 319–330, 2004.

4. G. De Micheli and L. Benini. *Networks on Chips*. Morgan Kaupmann, San Francisco, CA, 2006.
5. P. Dongkook, S. Eachempati, R. Das, A. K. Mishra, Y. Xie, N. Vijaykrishnan, and C. R. Das. Mira: A multi-layered on-chip interconnect router architecture. In *Proceedings of International Symposium on Computer Architecture*, pp. 251–261, 2008.
6. M. Gschwind, P. Hofstee, B. Flachs, M. Hopkins, Y. Watanabe, and T. Yamazaki. A compiler enabling and exploiting the cell broadband processor architecture. *IBM Systems Journal Special Issue on Online Game Technology*, 45(1), 2006.
7. R. Ho, K. Mai, and M. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490–504, April 2001.
8. A. Jantsch and H. Tenhunen. *Networks on Chip*. Kluwer Academic Publishers, Boston, 2003.
9. J.Kim, J. Balfour, and W. J. Dally. Flattened butterfly topology for onchip networks. In *Proceedings of International Symposium on Microarchitecture*, pp. 172–182, 2007.
10. J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, N. Vijaykrishnan, and C. Das. A novel dimensionally-decomposed router for on-chip communication in 3D architectures. In *Proceedings of International Symposium on Computer Architecture*, pp. 138–149, 2007.
11. J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. S. Yousif, and C. Das. A gracefully degrading and energy-efficient modular router architecture for on-chip networks. In *Proceedings of International Symposium on Computer Architecture*, pp. 4–15, 2006.
12. J. Kim, D. Park, C. Nicopoulos, N. Vijaykrishnan, and C. Das. Design and analysis of an NoC architecture from performance, reliability and energy perspective. In *Proceedings of Symposium on Architecture for Networking and Communications Systems*, pp. 173–182, October 2005.
13. P. Kongetira, K. Aingaran, and K. Olukotun. Niagara: A 32-way multithreaded SPARC processor. *IEEE MICRO*, 25(2):21–29, 2005.
14. F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir. Design and management of 3D chip multiprocessors using network-in-memory. In *Proceedings of International Symposium on Computer Architecture*, pp. 130–141, 2006.
15. G. H. Loh. 3D-stacked memory architectures for multi-core processors. In *Proceedings of International Symposium on Computer Architecture*, pp. 453–464, 2008.
16. I. Loi, F. Angiolini, and L. Benini. Developing mesochronous synchronizers to enable 3D NoCs. In *Proceedings of Design, Automation and Test in Europe Conference*, pp. 1414–1419, 2008.
17. I. Loi, S. Mitra, T. H. Lee, S. Fujita, and L. Benini. A low-overhead fault tolerance scheme for tsv-based 3D network on chip links. In *Proceedings of International Conference on Computer-Aided Design*, pp. 598–602, 2008.
18. P. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, and G. Hallberg. Simics: A full system simulation platform. *IEEE Computer*, 35(2):50–58, February 2002.
19. R. Marculescu, U. Y. Ogras, L. S. Peh, N. E. Jerger, and Y. Hoskote. Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(1):3–21, January 2009.
20. R. Mullins, A. West, and S. Moore. Low-latency virtual-channel routers for on-chip networks. In *Proceedings of International Symposium on Computer Architecture*, p. 188, June 2004.
21. V. F. Pavlidis and E. G. Friedman. 3-D topologies for networks-on-chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(10):1081–1090, 2007.
22. L. Peh and W. Dally. A delay model and speculative architecture for pipelined routers. In *Proceedings of International Symposium on High Performance Computer Architecture*, pp. 255–266, January 2001.
23. A. Pullini, F. Angiolini, S. Murali, D. Atienza, G. De Micheli, and L. Benini. Bringing NoCs to 65 nm. *IEEE Micro*, 27(5):75–85, 2007.
24. T. Richardson, C. Nicopoulos, D. Park, V. Narayanan, Y. Xie, C. Das, and V. Degalahal. A hybrid SoC interconnect with dynamic TDMA-based transaction-less buses and on-chip networks. In *Proceedings of International Symposium on VLSI Design*, pp. 657–664, 2006.
25. S. Rusu, S. Tam, H. Muljono, J. Stinson, D. Ayers, J. Chang, R. Varada, M. Ratta, and S.

Kottapalli. A 45 nm 8-core enterprise xeo processor. In *Proceedings of International Solid-State Circuits Conference*, February 2009.

26. J. Shen and M. Lipasti. *Modern Processor Design: Fundamentals of Superscalar Processors*. McGraw-Hill, Boston, 2005.
27. P. Shivakumar and N. Jouppi. Cacti 3.0: An integrated cache timing, power and area model. In *Technical Report, Compaq Computer Corporation*, August 2001.
28. Standard Performance Evaluation Corporation. SPEC OMP. http://www.spec.org.
29. G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen. A novel architecture of the 3D stacked mram l2 cache for CMPs. In *Proceedings of International Symposium on High Performance Computer Architecture*, pp. 239–249, 2009.
30. B. Vaidyanathan, W. Hung, F. Wang, Y. Xie, N. Vijaykrishnan, and M. Irwin. Architecting microprocessor components in 3D design space. In *Proceedings of International Conference on VLSI Design*, pp. 103–108, 2007.
31. S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar. An 80-tile sub-100-w teraFLOPS processor in 65-nm cmos. *IEEE Journal of Solid-State Circuits*, 43(1):29–41, 2008.
32. Y. Xie, G. H. Loh, B. Black, and K. Bernstein. Design space exploration for 3D architectures. *ACM Journal of Emerging Technology of Computer Systems*, 2(2):65–103, 2006.
33. Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang. A low-radix and low-diameter 3D interconnection network design. In *Proceedings of International Symoposium on High Performance Computer Architecture*, pp. 30–41, 2009.
34. S. Yan and B. Lin. Design of application-specific 3D networks-on-chip architectures. In *Proceedings of International Conference of Computer Design*, pp. 142–149, 2008.

第9章　PicoServer：使用三维（3D）堆叠技术建立能源效率服务器

Taeho Kgil，David Roberts，Trevor Mudge

由于电力和冷却技术的增长有助于降低数据中心的成本，能源效率成为驱动服务器设计的关键。提高能源利用效率的方法之一是采用创新互连技术，如3D堆叠。三维堆叠技术给未来获得低功耗、高集成和可移动性的服务器提供了新的机遇。本章介绍了一种叫作PicoServer的架构，它采用3D技术把一个含有一些简单低速的处理器内核芯片与满足主存储器要求的多存储器吸纳片键合在一起。这些多处理器芯片由DRAM组成。3D堆叠的使用有助于增加处理器和存储器之间的低延时的路径。这样便无须二级缓存，并使得其面积被重新分配到其他简单内核上。这些芯片可以在不损耗吞吐量的情况下降低时钟频率。较低的时钟频率意味着更容易满足3D堆叠的热约束。PicoServer相当简单，仅仅需要通过最简单的3D技术将芯片在彼此顶部堆叠。由于引入的新技术（3D）所实现的一系列低成本、低功耗和高集成服务器架构会产生一定的风险，我们的目的是将其降到最低。

9.1　介绍

数据中心是当今计算平台不可或缺的一部分。互联网和摩尔定律的持续发展已经使互联网服务提供商（如雅虎）建立起具有数百万台服务器的大型数据中心。对于大规模的数据中心，提高能源效率便成了其关键任务。基于现成的通用处理器的数据中心会消耗大量不必要的电能，且需要昂贵的冷却系统，并占据了很大的空间。实际上，功率和冷却这些数据中心的成本很可能会成为运行成本的一个重要部分。我们这种说法可以在图9-1中得到证实，这超出数据中心每年的经营成本。图9-1清楚地显示出，功耗与冷却系统成本的逐年增长使得整体数据中心的运行成本也随之增长。

能源效率服务器的设计方法之一是引进创新的互连技术。三维堆叠技术是一种互连技术，能显著提升新的单芯片多处理器架构的能源效率。我们提出的架构PicoSever㊀，采用了3D技术把一个含有一些简单低速处理内核的芯片与许多存储器芯片键合在一起，构成主存储器。另外，用3D堆叠实现存储处理器的互连，可

㊀　本章是基于文献［32］和文献［29］的工作。

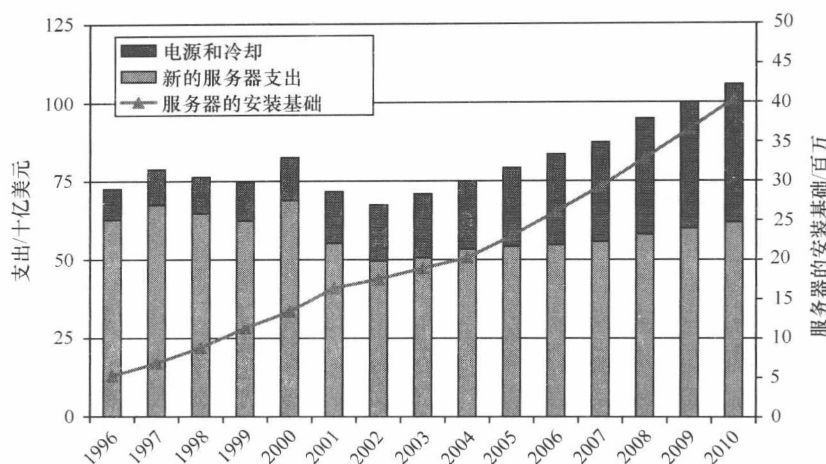


图 9-1　每年的 IDC 预计的功耗、冷却服务和购买其他服务的费用[52]

拥有非常高的带宽和相当低的延迟。因此省去了复杂的缓存结构。可以看出一般芯片上，将二级缓存的花费用于附加处理器内核的花费会更好。额外的内核意味着在吞吐量不变的情况下，运行得很慢。速度较慢的内核能让我们减少功率消耗和其带来的温度约束，而这些对于 3D 电路来说正是潜在的阻碍。所形成的系统非常适合于在吞吐量方面的应用，例如服务器。我们提出的架构相当简单，芯片仅需要最简单的 3D 技术堆叠在彼此顶部。我们的目的是将已实现的一系列低成本、低功耗和高集成服务器架构的风险降到最低。使用 PicoSever 能显著地满足低功耗和小空间的需求。

服务器应用程序在处理按基于每个客户端的事件时相互独立，并能展示高度线程级的并行性。这种高度并行，并不适合传统的单片式处理器。CMP 可由许多简单内核建立，当保持相似水平的吞吐量时，可在一个较低的频率上使用线程级并行，因此耗散功率较小。通过将用 3D 堆叠键合，我们能展示出其具有进一步降低功耗需求的可能。三维堆叠可实现以下关键改进：

1）在 DRAM 和一级缓存之间其支持多个内核的高带宽总线，即具有数以千计的低延迟连接线，且芯片间实现最小面积是有可能的。因为互连总线是在片上的，所以相对于在芯片间实现来说，能够以相对低的功耗预算，使用更宽的总线。

2）由于大容量片上的 DRAM 集成，需要在存储器层次结构上修改。移除二级缓存并由更多的处理器内核代替是可能的。因为地址复用和片外 I/O 驱动焊盘[47]不再需要，相应的片上的 DRAM㊀存取延迟也减少了。此外，它也带来一种具有快速的片上 DRAM 来建立非均匀存储结构和相对较慢的片外二级系统存储器的机会。

㊀　我们参照堆叠在主服务芯片上的芯片作为片上，这是因为它们能形成 3D 芯片。

3）系统功耗整体减小主要由于内核时钟频率的减小。在 1）和 2）的陈述中 3D 堆叠的好处是让多核心的主频集成在一个适度的片上频率，即工作在 500 ~ 1000MHz 之间，并同时提供高的吞吐量。减小内核时钟频率可以简化它们的结构，例如通过使用较短的流水线来降低向前逻辑。

3D 堆叠的潜在劣势是受热限制（见第 3 章）。然而，对于我们在 9.4.5 节提出的简单、低功耗内核类型的 PicoServer 来说，这并不是一种限制。事实上，根据表 9-2 中 ITRS 预测，可预测出系统仅消耗几瓦特，甚至不需要散热器。

PicoSever 的总体架构如图 9-2 所示。为了本次的工作目标，我们假设一个 5 ~ 9 层芯片的堆叠。它们的连接是通过垂直于芯片上的通孔实现的。3D 互连通孔的尺寸在 1 ~ 3μm 之间，具有 1 ~ 6μm 之间的间距。当今的商业能支持每平方厘米 1000000 个通孔[26]。这远远超过了 PicoSever 需要的个数。这些通孔可作互连和热通道。在研究中，我们假设基于组件逻辑的微处理器内核，网络接口控制器（NIC）和外围设备都是处于底层而以面向常规容量的 DRAM 占据其他层。为了理解设计空间和这项新技术的潜在的优势，我们通过仿真探索了不同的总线宽度、内核数量、频率和存储结构之间的折中关系。我们发现对于架构具有两个 250MHz 时钟周期的延时的 1024 位的总线宽度是合理的。另外，按合理的面积预算约束，在 90nm 工艺下，芯片面积保持在 $80mm^2$ 以下。在最大芯片上，12 核 PicoSever 结构所占据的面积保守估计为 $80mm^2$。芯片上 4 核和 8 核的 PicoServer 结构的面积分别是 $40mm^2$ 和 $60mm^2$。

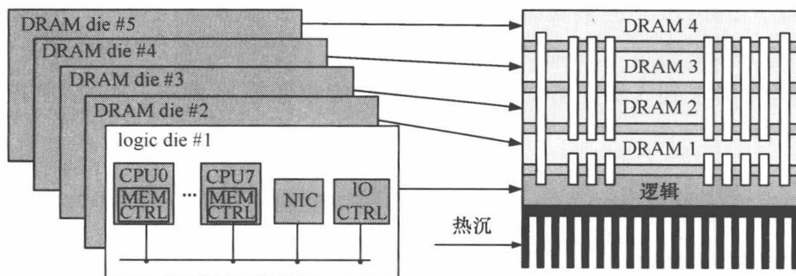


图 9-2　PicoSever 的描述图：采用 3D 堆叠技术将 CMP 架构连接到 DRAM 上并具有片上网络接口控制器（NIC），来提供低延迟高带宽网络

我们也拓展了 PicoSever 的分析并展示了在 PicoSever 架构上集成 Flash 的影响。对于这两种配置，我们提供了定性分析：①把 Flash 作为分立元件；②在 DRAM 和逻辑的堆叠之上直接堆叠 Flash。第一种结构由更大的系统内存容量驱动，而第二种结构的驱动可为小型系统。

本章安排如下：在 9.2 节我们通过描述服务器平台的概况、3D 堆叠技术和 DRAM 技术的趋势来说明此次工作的背景。在 9.3 节，我们概述了设计空间探索的方法。在 9.4 节，我们提出了 PicoServer 架构的更多细节，并评估了各种

PicoServer 结构。在 9.5 节，我们展示出了有 PicoServer 服务器架构的基准测试与传统架构且没使用 3D 堆叠技术相对比的结果。这些架构是没有 3D 堆叠的 CMPs 和常规的具有奔腾 4 特性的高性能桌面架构。摘要和总结性的评论在 9.6 节中给出。

9.2 背景

本节提供了目前的服务器平台、3D 堆叠技术和 DRAM 技术的状态概况。首先展示了目前服务器是如何在数据中心被部署的，并且分析了当前服务器的负载表现。接下来，解释了 3D 堆叠技术的状态和它在这项工作中是如何被应用的。最终，展示出了 DRAM 技术的优点，并解释了 DRAM 在服务器领域的目前和将来的趋势。

9.2.1 服务器平台

9.2.1.1 三层服务器体系结构

现在数据中心普遍是围绕一个 3 层架构的服务器建立的。图 9-3 展示了一个 3 层服务器结构和它是如何处理一个服务请求的。第 1 个层次处理大批量的客户端（客户终端）的请求。第 1 层级服务器处理 Web 请求。因为第 1 层级服务器基于每个客户端上处理事件，它们是独立的并显示高级别线程并行。对于需要大量的计算和数据库访问的请求，会被转发到 2 级服务器。2 级服务器执行用户应用程序，该程序能够翻译脚本语言，并确定哪些对象（通常数据库对象）应被访问。2 级服务器生成数据库请求到 3 级服务器。3 级服务器接收数据库请求，并将结果返回到第 2 层的服务器。

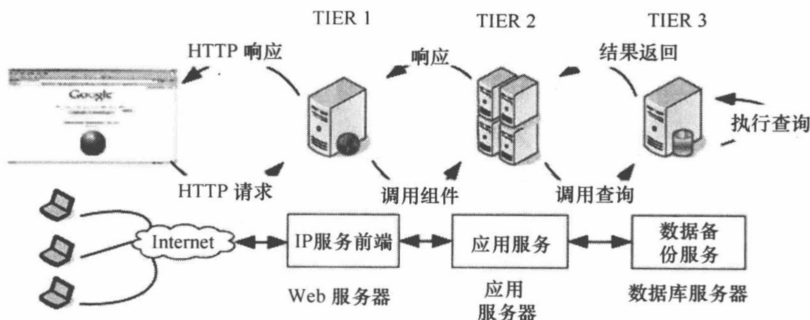


图 9-3 一个典型的 3 层服务器架构。第 1 层——Web 服务器，第 2 层——应用服务器，第 3 层——数据库服务器

例如，当一个客户请求 Java Servlet Page（JSP 网页）时，它将被前端的 1 级服

务器接收到。1级服务器识别和处理Java Servlet Page，并启动通常使用远程接口信息（RMI）向2级服务器发起请求。2级服务器在3级服务器上发起数据库请求，即返回产生的结果并将相关的信息发送到这个流程上，一直到1级服务器。最终，1级服务器将生成的内容发送给客户端。

3级服务器架构普遍用于当今的服务器网络中，因为他们允许每个级别对其负载进行优化。然而，这种策略并不总是被采用。谷歌公司在每个级别上都采用了相同的服务器，因为此规模和管理问题的代价是弊大于利的。除了三级服务器的数据库磁盘系统，我们将展示，通用Picoserver架构适合所有层次。

9.2.1.2 服务器负载特性

服务器的工作负载显示了高的线程级并行性（TLP），因为通过连接级并行性，客户端的连接可以容易地映射到线程级并行。表9-1显示了商业服务器负载的表现。除了决策支持系统，大多数商业服务器的工作负载显示了高的线程级并行性和低的指令级并行性（ILP）。然而，为了利用ILP，常规的通用处理器通常被优化。这些工作负载受高缓存未命中率的影响经常暂停机器。这导致每个周期的指令低且处理器资源利用率差。我们的研究说明了这一点，除了对于计算大量的工作负载的情况，像PHP应用服务、视频流服务和决策支持服务，对于典型的服务负载，乱序的处理器具有0.21~0.54之间的IPC值，例如，具有2MB的二级缓存的最合适的计算负载。这些工作负载的表现并不好是因为许多请求数据已经被直接从磁盘存取到内存系统，无效的缓存数据导致了缓存数据的未命中。因此，我们大致可以说，单线程优化的乱序处理器在服务器上的工作负载表现得并不良好。另一个令人惊讶的是大多数服务器负载将相当大的时间花费在内核代码上，而SPECPU基准则不是这样。这个内核代码主要涉及中断处理的网络接口控制器（NIC）或磁盘驱动器、数据包传输、网络堆栈处理和磁盘缓存处理。

表9-1 文献［38］中适应商业工作负载的行为

属性	Web99	JBOB（JBB）	TPC-C	SAP 2T	SAP 3T DB	TPC-H
应用类别	网页服务	Java服务	OLTP①	ERP②	ERP	DSS③
指令集并行	低	低	低	中等	低	高
线程级并行	高	高	高	高	高	高
指令/数据的工作集	大规模	大规模	大规模	中等	大规模	大规模
数据共享	低	中等	高	中等	高	中等
高I/O带宽	高（网络）	低	高（盘）	中等（盘）	高（盘）	中等（盘）

① OLTP：联机事务处理。

② ERP：企业资源规划。

③ DSS：决策支持系统。

最终，申请的很大一部分都围绕着同组文件。这些文件的访问请求接入内存和I/O。由于适度的计算需求，存储器和 I/O 的延迟是高性能的关键。因此，在系统内存中磁盘缓存对提供足够的吞吐量起到了重要的作用。若无磁盘缓存，因硬盘驱动延迟带来的性能下降是不可接受的。

若使这类工作负载表现良好，此架构应自然地支持多线程来响应众客户端的独立请求。这种情况表明，CMP 或 SMT 架构要更好地利用处理器芯片面积。

9.2.1.3　常规服务器的功耗组成

图 9-4 展示出当今的服务器平台的功率下降。此服务器使用多处理器芯片和简单有序的内核来实现功耗的降低。功率下降表明其中 1/4 是由处理器消耗的，1/4 是由系统内存消耗的，1/4 是由电力供应消耗的，1/5 是由 I/O 接口消耗的。由此可以看出，大量使用系统内存会导致大幅度的功率消耗。这将随着系统内存时钟频率和内存大小的增加而增加。我们还发现，尽管使用较简单的高能源利用率的内核，处理器仍然会产生大量功耗。由于片外接口需要高的 I/O 供电电压，因此 I/O 接口产生了大量功耗。I/O 的供电电压很有可能随以后的规模而降低，但是可能不会如内核供电电压的规模那样快速降低。因此有机会通过片上集成系统组件进一步减小功耗。最后，我们发现根据电源供应的显示，这效率有些低下。这是因为它不得不支持多级电压。减少电源轨数量将极大地提高电源效率。三维堆叠技术的潜在的好处有：①通过芯片上集成更多的系统组件减少由处理器和 I/O 接口消耗的功率；②通过在服务器中减少电源轨的数目来提高电源效率。

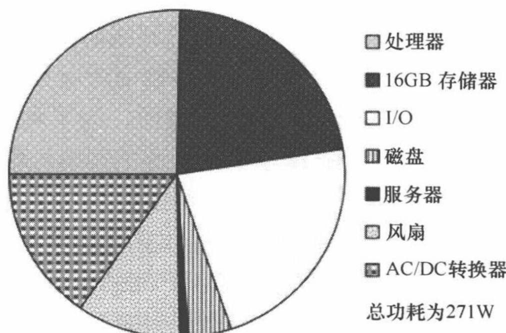


图 9-4　T2000 UltraSPARC 执行 SPECJBB 的功率下降

9.2.2　三维堆叠技术

这部分提供了 3D 堆叠技术的概况。在过去，学术上和工业上都付出了大量的努力来实现 3D 堆叠技术[17,40,37,44,57]，但成败参半。这是因为需要解决许多挑战。它们包括：①实现高产量的键合芯片堆叠；②给每个叠层提供电源；③控制芯片堆叠引起的局部热效应。然而，在过去的几年中强大的市场力量促进了对移动终端领域的小型化及超低功耗的需求。对此，多个商业企业已经开始提供可靠的低成本芯片到芯片的 3D 堆叠技术。

在 3D 堆叠技术中，典型的芯片键合方式是面对面或者面对背。相对于面对背的键合方式，面对面的键合方式提供了较高的芯片到芯片的通孔密度和更低的面积

消耗。面对背较低的通孔密度是因为硅通孔（TSV）不得不通过硅衬底。图 9-5 展示出如何使用 3D 堆叠技术堆叠芯片的高层次例子。第一层（从底部开始算起的）和第二层之间是面对面的键合，而第二层与第三层之间是面对背键合。在 3D 堆叠技术中使用键合技术，开启了异构芯片堆叠在一起的契机。例如，堆叠 DRAM 和逻辑单元的结构是由不同的工艺步骤制造的。本章参考文献[43，24，16]说明了在逻辑上堆叠 DRAM 的好处。另外，在垂直方向上增加第三个维度，整体互连的线长减小，且可在较低成本的面积上增大总线宽带。相比全局互连，此 3D 通孔的寄生电容和电阻是可以忽略不计的。我们还注意到，对于 3D 通孔的尺寸和间距，仅附加了一个可接受的面积开销。在 90nm 技术下，三维通孔间距相当于 22λ，这大约是一个 6T SRAM 单元的尺寸。随着技术的成熟，该尺寸将不断地缩小。

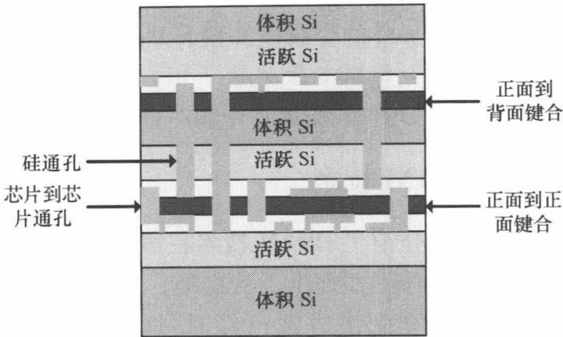


图 9-5 一个 3 层 3D IC 的例子

在表 9-2 中的 ITRS 发展路线图预测了在不久的将来会进一步使用堆叠技术。这些连接是通过垂直于芯片上的通孔。正如前面提到过的，3D 互连通孔的尺寸在 1～3μm 之间，并有 1～6μm 的间距。目前商业上能支持每平方厘米 1000000 个通孔[26]。

表 9-2 对于功耗感知平台，ITRS 预测[12]了 3D 堆叠技术、存储器阵列单元和最大的功耗预算。相比于 SRAM，ITRS 预测表明当 DRAM 密度超过 SRAM 密度 15～18 倍时，将会采用 3D 堆叠技术将大容量 DRAM 集成在芯片上

	2007 年	2009 年	2011 年	2013 年	2015 年
低成本/手持 #芯片/堆叠	7	9	11	13	14
SRAM 浓度/(Mbit/cm^2)	138	225	365	589	948
在生产时 DRAM 浓度/(Mbit/cm^2)	1940	3660	5820	9230	14650
最大功率预算性价比系统/W	104	116	119	137	137
最大功率预算的低成本/手持系统与电池/W	3.0	3.0	3.0	3.0	3.0

表 9-3 三维堆叠技术参数[26,13,44]

	正面到背面	面对面	RPI	MIT 3D FPGA
大小	1.2μ×1.2μ	1.7μ×1.7μ	2μ×2μ	1μ×1μ
最小距离	<4μ	2.4μ	N/A	N/A
通过电容	2～3fF	≈0	N/A	2.7fF
串联电阻	<0.35Ω	≈0	≈0	≈0

使用3D堆叠技术的整体产量是由每个单独的芯片层的产量决定的。因此，致力于设计具有高产量的芯片块是重要的。存储器堆叠相比于逻辑到逻辑的堆叠是更好的选择。通常存储器件具有更高的产量，因为容错正好符合它们的重复结构。例如，重新融合额外的位线以补偿有缺陷单元和应用单比特纠错逻辑来提升存储器的产量。一些研究包括本章文献［48］显示 DRAM 产量极高，这意味着建立单层逻辑芯片和多层 DRAM 堆叠芯片的产量会接近普通的逻辑芯片的产量。

9.2.3 DRAM 技术

这一部分介绍了 DRAM 技术在服务器空间的优势。今天的 DRAM 通常由应用空间确定的各种形式提供。特别地，对于服务器平台，DDR2/DDR3 DRAM 作为存储器系统的主要解决方案已经时过境迁。相比于 DDR2/DDR3，FBDIMM DRAM 具有更高的吞吐量和更高的容量，已作为一个替代产品出现，但其功耗大的问题仍有待解决。而网络负载的服务器空间 RLDRAM 和 NetRAM[55,7]也是很受欢迎的选择。

这些存储器的普通特性是高吞吐量和低延时。在服务器空间，DRAM 必须满足高吞吐量和低延时来实现高性能。但是只有增加 DRAM I/O 接口和 DRAM 阵列的功耗才能达到高性能。因此 I/O 和 DRAM 功耗占据整体功耗的显著部分（正如我们在 9.2.1.3 节中所提到的)。工业上通过减少 I/O 电源电压和引进以牺牲吞吐量和延迟为代价的低功耗版本的 DDR2 接口来解决这个问题。我们将证明，使用 3D 堆叠技术的 DRAM 堆叠可以提供高吞吐量和低延时 DRAM 接口的同时具有更小的功耗。

9.3 方法

这一部分描述了我们在评估 3D 堆叠技术好处时使用的方法。我们研究的架构是从一个叫作 M5[15]的微架构模拟器获得，M5 能够运行 Linux，并评估全系统级性能。通过全系统模拟器来模拟 3D 堆叠技术带来的好处，同时模拟了在 M5 上连接到多个客户端的多个服务器，客户端请求由用户级网络应用程序生成。通过衡量服务器的吞吐量，即网络带宽或每秒传输量可以估计其性能。芯片面积估算源于以前的出版物，延时和功耗通过已开发的模型来估算[12,26,50,2,13,17]。DRAM 时序和功率从 IBM 和 Micron 技术数据表［3］中获得。我们的方法将在下面的章节中详细描述。

9.3.1 仿真研究

9.3.1.1 全系统架构仿真

为了估算 PicoSever 的性能，使用了 M5 全系统仿真。在可配置的架构上，M5

启动未修改的 Linux 内核。在模拟器中多个系统被定义，用来模拟客户端和服务器，并通过以太网链路模型连接。在服务器端可执行 Apache-网络服务器，凤凰—视频流媒体服务器，mySQL—数据库服务器和 NFS—文件服务器。在客户端可执行代表请求动态及静态网页内容的基准程序、视频流请求、数据库查询以及网络连接文件。为了便于比较，我们定义奔腾 4 系统[53]和类似多处理器系统芯片，这与本章参考文献［36］中所示类似。另外介绍了在这些平台上使用 3D 堆叠技术的几种配置。我们假设用 3D 堆叠技术，那么更宽的总线宽度可在低功耗下实现。表 9-4 显示了所使用的模拟器的配置。

表 9-4 通用仿真配置。DDR2 DRAM 模块中产生的系统存储器延时。我们假设核时钟频率（低于 1GHz）有更高的缓关联性。由于在多核平台较长的全球互连长度[39]，L2 缓存卸载延迟对于单核和多核的配置不同

	OO4 小型基准与带和不带 3D 堆叠	OO4 大型基准与带和不带 3D 堆叠	常规的 CMP MP4/8 不带 3D 堆叠	PicoMP4/8/12-500MHz/1000MHz①
工作频率	4GHz	4GHz	1GHz	500MHz/1GHz
处理器数量	1	1	4/8	4/8/12
处理器类型	out-of-order	out-of-order	in-order	in-order
发射宽度	4	4	1	1
一级缓存	2way 16KB	2way 128KB	4way 16KB per core	4way 16KB per core
二级缓存	8way 256KB 7.5ns unloaded latency	8way 2MB 7.5ns unloaded latency	8way 2MB 16ns unloaded latency	N/A
内存总线宽度	64bit @400MHz/ 1024bit@250MHz	64bit @400MHz/ 1024bit@250MHz	65bit@250MHz	1024bit@250MHz
系统内存	512MB DDR2 DRAM	512MB DDR2 DRAM	512MB DDR2 DRAM	128MB - 512MB DDR2 DRAM

① PicoServer 平台上使用 3D 堆叠技术。PicoServer 的核心时钟频率通常 500MHz。与 1GHz 的内核时钟频率 PicoServer 配置以后用于显示 3D 堆叠技术的影响。

9.3.1.2 服务器基准测试

我们使用几种能和客户请求直接联系的基准测试。为了测量网络服务的性能，使用两种网络内容处理基准测试即 SURGE[14]和 SPECweb99[10]。这两个基准测试要求超过 1GB 的文件集。一个使用 PHP 的网页脚本处理基准测试 SPECweb2005[9]被选择来表示脚本工作负载。一种使用 RTSP 协议和 UDP 协议的视频流基准测试 Fenice[6]被选择来测量响应试工作负载的表现。对于一个文件共享基准测试，使用 NFS 服务器并用 dbench（基准测试）强调它。最后，我们进行了两个数据库的基准测试来衡量第二层和第三层工作负载的数据库性能。

SURGE。SURGE 基准测试代表客户对静态网页内容的需求。我们修改 SURGE

的文件集并使用一个 zipf 分布产生合理的客户端请求。基于 zipf 分布，请求一个大约 12KB 的文件大小的静态网页，将花费客户请求时间的 50%。

SPECweb99。为了评估一个混合的静态网站和简单的动态网站内容，我们使用一个 SURGE 的修改版本来请求 SPECweb99 文件集（其表现在表 9-1 中被说明）。对于 SPECweb99，我们使用默认的配置来产生用户请求。70% 的客户请求是对于静态网站内容的，30% 的客户请求是对于动态网站内容的。

SPECweb2005。脚本语言是一种描述网页的流行的方式。SPECweb2005 提供 3 种类型的基准：银行基准——模拟用户的网上银行活动，电子商务基准——模拟网上购买活动，支持基准——模拟网上流活动。所有的基准测试需要一个脚本解释器来产生动态网页。我们使用 PHP 解释器来测量 2 级服务器的行为。从 SPECweb99 和 SURGE 客户方法的描述生成的客户端请求。

Fenice。视频请求服务也是 1 级服务器的一个重要的工作负载。对于版权保护和实况广播来说，RTSP 协议通常用于实时视频播放。Fenice 是一个开源项目流[6]，它提供工作负载并支持 RTSP 协议。我们修改它来支持多线程。客户请求由一个 nemsi 的修改版本和一个支持 MPEG 播放器的 RTSP 生成。Nemesi 也来自于开源项目流工程。我们产生了能够充分利用服务器 CPU 720 ×480 分辨率的 MPEG2 帧的高质量 16Mbit/s 的数据流的多个客户端请求。

dbench。这个基准测试通常用来强调 NFS 守护进程。在测试中，我们使用多线程的且在 Linux 内核中可用的 NFS 守护进程。客户端用 dbench 产生 NFS 通信，已锁定文件服务器。Dbench 产生读写文件服务器的工作负载的同时锁定这些文件使得不同的客户端不能同时访问。

OLTP。在线交易处理是一个典型的在 2 级和 3 级服务器上执行的工作负载（其行为在表 9-1 中示出）。TPC 委员会对 OLTP 基准进行了详细的描述。我们使用 TPC-C 的修改版本由称为 DBT2[5] 的开放源代码开发实验室（OSDL）提供。DBT2 生成交易订单。我们的数据库服务是 MySQL 5.0。我们使用 InnoDB 存储引擎、支持交易并对多核提供合理数量的扩展。我们产生了 1GB 仓库，其通常用于计算小规模密集型数据库。由于仿真时间限制，我们选择了一个小的工作集大小。据此我们选择缓冲池的大小。

DSS。决策支持系统是另一个典型的工作负载，它用于评估第二层和第三层服务器。我们使用 TPC-H，即 DSS 工作负载的当前版本。由 OSDL（DBT3）[5] 提供的 TPC-H 的再修改版本在本研究中被使用。我们加载了 TPC-H 数据库到 mySQL 数据库系统，并且使用确定的 TPC-H 查询来衡量性能。查询缓存被禁用，以防止由于缓存带来的查询时间加速。为了将模拟时间缩短到合理的量，在许多 TPC-H 查询中我们只进行和测量 Q22 查询的时间。Q22 查询花费适量的时间来执行和说明其性能，见表 9-1。

9.3.2 估算功率及面积

功耗和面积的估算很难做到非常精确。为了做出合理的估算和展示总体趋势，我们采纳了行业白皮书、数据表和在芯片面积学术界的出版物，比较了最初分析的功耗模型与实际的实施结果，并广泛使用了周期级仿真技术。在接下来的小节中，我们将对此进行进一步讨论。

9.3.2.1 处理器

我们很大程度上依赖于用 ARM 处理器的数据报道[20,1,51]来估算处理器的功耗和芯片面积。ARM 是一个简单的顺序 32 位处理器代表，它是适合于 PicoServer 的。由于和 PicoServer 核架构的相似之处，根据本章参考文献［20，1，51］发表的数据，我们推断核心功耗为 500MHz 的 PicoServer 的芯片面积和功耗。表 9-5 列出这些估计值同时在文献［1，51］中也列出，并和奔腾 4 核的值做对比。还对每个核心芯片的预期面积进行了分析。我们从 ARM、MIPS、PowerPC 和其他类似规模的顺序处理器那里收集了几个处理器芯片的面积数。

表 9-5　各种微处理器发布的功耗值

	奔腾 490nm	ARM11 130nm	Xscale 90nm①	PicoServer MP 90nm②
一级缓存	16KB	16KB	32KB	16KB
二级缓存	1MB	N/A	N/A	N/A
总体功耗/W	89 ~ 103W	250mW@550MHz	850mW@1.5GHz	190mW@500MHz
总体芯片面积/mm²	112	5 ~ 6	6 ~ 7	4 ~ 5

① 对于 90nm 的不包括二级缓存的 Xscale 芯片面积[51]。

② 对于 PicoServer 核，我们估计功耗在 ARM11，Xscale 的范围内。

表 9-5 中所列的功率值包括静态功耗。与 ARM 核的值相比，我们对于 500MHz 的 PicoSever 核的估算是保守的，尤其是对于本章参考文献［51］。鉴于 Xscale 核在 1.5GHz 和 1.3V 下功率为 850mW，当施加 3 倍时钟频率和有额外扩大电压的机会时，对于 90nm 工艺的 500MHz 频率的 PicoServer 核，预计功率消耗值为 190mW（毫瓦）是保守的。对于在其他核时钟频率，如 1GHz，我们产生电源与频率图。它遵循立方关系[23]。我们假设了一个逻辑深度为 24 FO4（扇出 4）的逻辑门，并且使用 90nm 工艺 PTM 技术[51]。

未来在 PicoServer 核中支持 64 位似乎不可避免。当我们看到了像 MIPS 和 Xeon 这样商用的内核为了支持 64 位而造成的面积和功耗的开销之后，对于 PicoSeve 核，我们预期，因支持 64 位造成的面积和功耗的开销是适中的。对于二级缓存，我们参考了本章文献［56］，并从实际的测量中缩放了面积和产生的功耗。假设当缓存存取率是 100% 时，本章文献［56］中的功率值产生。因此，当假设漏电流消耗二级缓存总功率的 30% 时，我们通过大小和存取率衡量二级缓存。

9.3.2.2　考虑三维堆叠技术的互连

为了研究的目的，我们采用了本章参考文献［12，26，50］中公布的数据作为典型的3D堆叠互连。一般来说，芯片到芯片的连接电容小于3fF，我们通过从3DMagic建模中提取的寄生电容，也验证了这一点。3DMagic是由MIT近期开发的工具。我们发现提取的电容值是2.7fF，符合本章参考文献［26］中的结果。通过与二维片上互连的比较，一个全局具有每mm 400fF电容的互连线被建立，这些基于本章参考文献［27］。因此，我们假设在3D互连中起到连接作用通孔的电容是可以忽略的。对于可能在芯片之间的I/O互连的数量，本章参考文献［26］报道了每mm^2 10000个的连接数值。我们需要的没有这么多。根据研究，我们需要大约1100个I/O连接：32位用于地址总线，1024位用于数据总线和其他的一些控制信号。为了估算在处理器和外围层中的互连电容，我们再次参考本章参考文献［27］来产生分析和建模值。我们选择了12mm的线长以满足$80mm^2$芯片，1.3倍的宽度/高度比，并且根据更小的芯片尺寸按比例缩小导线长度。与本章参考文献［22］所示的已有实施了的2D芯片的线上互连导线长度减少的预测相比，假设我们能减少33%的互连线电容。基于这些最初的值，我们从Hspice仿真中，计算所需的中继器数量来驱动在250~400MHz范围内的互连。由于在此范围内频率是相对较低的，我们只需要最多2~3个中继器来驱动这个线路。

我们测定触发速率以及电线接入速率，并使用公知的动态功率方程计算互连功率。表9-6出示了在2D的情况下，3D堆叠和2D芯片外实现预期的1024位的互连电容。粗略地讲，片上实施的电容最多为片外实施的33%。另外，因为在I/O端口处的供电电压通常是1.8~2.5V，高于通常芯片的供电电压，与片上相比，片外实施具有更高数量级的整体的互连功率功耗。我们在基准电路中发现，其具有适中的触发率，经典小配置适中的存取速率，并且适中的总线频率——250MHz，因此我们得出结论，芯片块（管芯）间的互连功率对整体功耗贡献极小。

表9-6　2D、3D和2D片外的1024位总线的电路的寄生互连电容

	130nm	90nm
片上2D 12mm	5.6nF	5.4nF
片上3D 8mm	3.7nF	3.6nF
片外2D	16.6nF	16.6nF

9.3.2.3　DRAM

我们使用本章参考文献［45］的数据作为PicoServer的DRAM面积估计。目前，合理地说，在90nm工艺节点下，64MB的DRAM芯片需要$80mm^2$面积。

常规DRAM与处理器是分开封装的，并通过I/O端口和线在PCB上访问。然而，对于我们的体系结构，DRAM存在于片上，并通过3D堆叠通孔连接到处理器和外围设备。因此，在设计中避免了由封装所必需的整个印制电路板的驱动片外信

号产生的功耗。使用 Micron DRAM 的电子表格计算器[3]，通过修正来省略热焊点功耗（端口），并从 M5 获得数据，这些数据包括花费在 DRAM 读取、写入和页面命中率上的周期数，进而得到 DRAM 的平均功率。我们比较了引自参考文献的 DRAM 估计功率，特别是与 SunFire T2000 服务器电源计算器[11]产生的 DRAM 功率值比较。Micron 电子表格使用实际的电流测量值对于每个 DRAM 操作——读、写、刷新、块预充等。我们假设了一个具有 1.8V 电源电压的设计。

9.3.2.4　网络接口控制器——NIC

由于缺乏对商业网络接口控制器（NIC）结构的详细信息，导致难以建立 NIC 功率的解析模型。在仿真中，我们以美国国家半导体公司的 82830 千兆以太网控制器作为参照。该芯片使用千兆位媒体独立接口（GMII）作为 MAC 层以太网卡和物理层（PHY）之间的接口。通过分析数据手册发现该芯片消耗的最大功率为 743mW[4]。这个功率基于 180nm 工艺。假定当所有的输入和输出引脚开启时的功耗为最大功耗，然后对数据降额以作为我们测试使用。此外，假定静态功率占芯片最大功耗的 30%。考虑到自本章参考文献［4］发表以来网络接口控制器（NIC）的显著发展，我们认为这个功耗模型是足够保守的。

9.4　PicoSever 架构

表 9-7 展示了常规 DRAM、XDR DRAM、二级缓存和应用 3D 叠堆技术的片上 DRAM 间延迟和带宽的对比。具备 1024 位带宽总线的 3D 堆叠片上 DRAM 在延迟和带宽上可与二级缓存、XDR DRAM 相媲美。这表明如果使用叠堆技术，则不需要二级缓存。此外，移除常规的 DRAM 片外驱动器可以减少 50% 以上[47]的访问延迟。这有力地证明了片上 DRAM 可作为有效的二级缓存使用。另一个佐证是，DRAM 厂商正在生产和推广减少随机存取延迟的 DRAM。因此，我们的 PicoServer 架构不具有二级缓存，而是使用片上 DRAM 通过一个共享总线体系结构连到每个核心中的一级缓存。此片上 DRAM 的作用是一个主系统存储器。

表 9-7　带宽和延迟表明相比在本章参考文献［39，56］中指出的二级缓存，片上 DRAM 可以方便地提供足够的内存带宽。SDRAM 和 DDR2 DRAM 的平均存取延迟估计为 *t*RCD + *t*CAS，其中 *t*RCD 表示 RAS 到 CAS 延迟，*t*CAS 表示 CAS 延迟。对于 XDRAM 而言，*t*RAC-R 被用来表示其延时，其中，*t*RAC-R 表示读取访问时间

	SDRAM	DDR2 DRAM	XDR DRAM	二级缓存 @1.2GHz	片上 DRAM 3D IC
带宽/(GB/s)	1.0	5.2	31.3	21.9	31.3
平均存取延时/(ns)	30	25	28	16	25①

① 不带 3D 堆叠感知优化平均访问延迟。当 3D 堆叠优化应用时，片上的 DRAM 延迟预计减少超过 50%[47]。

PicoServer 架构由单一问题的循序处理器组成一个多处理器芯片，这种结构天然适用于高水平 TLP 应用[36]。每个 PicoServer 的 CPU 内核有额定 500MHz 时钟，并且具有指令和数据缓存，同时数据缓存使用 MESI 缓存一致性协议。研究显示，大多数总线的流量是由缓存未命中流量产生的，而不是由于缓存一致性。这是因为目标应用程序空间的特性和较小的一级缓存，即 16KB 的指令和每个核心 16KB 的数据缓存。当前的密度下，在 PicoServer 的片上的 DRAM 堆栈的容量是几百兆字节。在不久的将来，这将上升到数千兆字节，如表 9-2 中所述。其他部分如网络接口控制器（NIC）、DMA 控制器和一个完整系统所需的附加外围设备都将被集成在 CPU 芯片上。

9.4.1 核心架构和多线程的影响

PicoServer 是由具有 5 级流水线的简单单问题循序内核组成。假设每个内核是 32 位架构。在一个服务器工作负载中分支预测仍然是有益的。每个核心拥有具有 1KB 历史表的混合分支预测器。研究表明，为服务器工作负载分支预测的准确度约为 95%。

每个核心还包括支持共享内存协议和存储器控制器直连 DRAM 的架构。存储器控制器对总线监听和缓存未命中做出反常响应。对于一个 DRAM 请求，存储器控制器传递地址、数据写入存储器或读取 CPU 的 ID。CPU 的 ID 需要用以返回读出数据的线路。估计单核芯片面积为 4 ~ 5mm^2（在表 9-5 示出）。

尽管可以从多线程得到一些好处（在后面的段落中描述），但由于仿真环境的限制我们假设不支持多线程。在不大量修改一个商用 Linux 内核的情况下，超过 16 个核或线程时很难按比例进行服务器应用。出于这个原因，我们在单核上运行多个线程以考察多线程。并以此推论在多核情况下，使用 3D 堆叠技术时的最佳线程数。多线程具有在长的位到内存时通过线程上下文切换来提高整体吞吐量的潜力。

为了研究对多线程 PicoServer 的影响，我们假设多线程支持包含一个完整的线程上下文，即寄存器文件、存储缓冲器和中断陷阱单元。需要一个额外的流水线级来调度线程。我们假定支持 4 个线程的芯片面积成本约为 50%。尽管本章参考文献［39］中预计 Niagara 核只需 20% 的芯片面积成本来支持 4 线程，然而我们的核要小得多——5mm^2 与 16mm^2。本章参考文献［20，51］估算寄存器和架构状态芯片面积占总芯片面积的比例更大。因此，对于 PicoServer 核我们估计了更大的面积成本。

在多线程研究中，我们改变从单核到存储器的可支持线程数和访问延迟以测量网络带宽（一种吞吐量指标）。我们分析了正在运行的 SURGE，因为其具有最高的一级缓存未命中率，这意味着它将从多线程中受益最多。在本研究中使用的指标是网络总带宽和每平方毫米网络带宽，并通过改变缓存大小来观察对线程的影响。

图 9-6 和 9-7 显示了模拟结果。从这些能够得出结论：线程确实有助于提高整

体吞吐量，但是考虑面积成本和 3D 堆叠的影响时这种提高只有有限的程度。三维堆叠通过简化存储器接口核心并降低传送等待时间的方式减少了存储器访问时间。图 9-6b 和图 9-7b 中所示的曲线三维堆叠存储器可以被访问数十次。后图表明，如果将面积效率和吞吐率放在一起考虑，限制到只有两个线程时最佳。我们还发现，增加核中额外的线程，内存和 I/O 流量同时增加。因此，系统必须能够提供足够的 I/O 和存储器带宽以适应增加的线程。否则，线程将不利于整个系统的吞吐量。

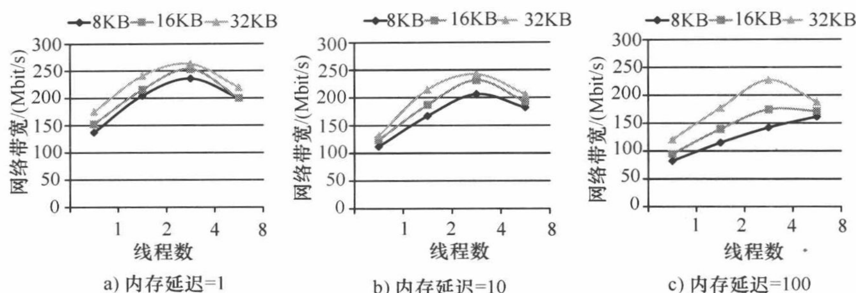


图 9-6 对于 SURGE，多线程的不同存储器延迟对于不同的 4 种方式设置相关联的缓存大小（8KB、16KB、32KB）和不同的线程数的影响。我们假设的核心主频为 500MHz

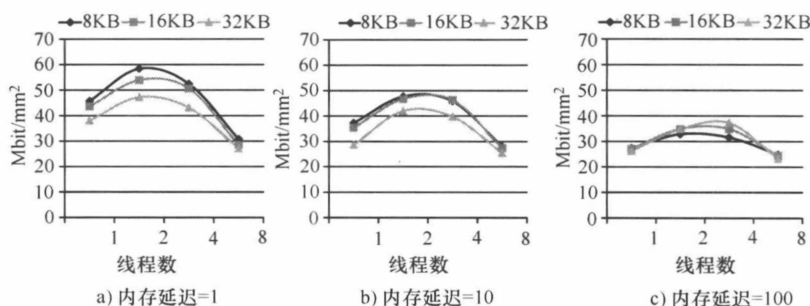
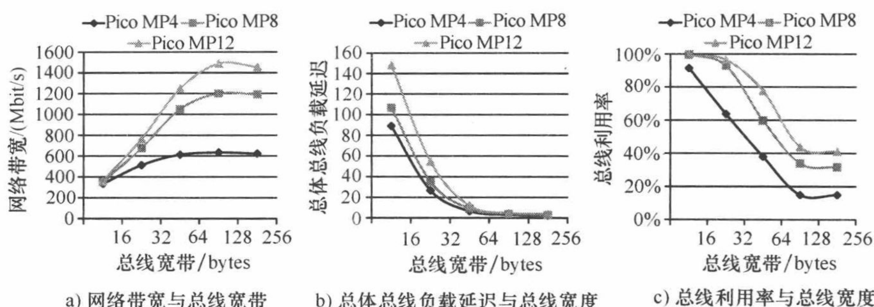


图 9-7 当存储器延时在 SURGE 中变化，多线程对于 $Mbps/mm^2$ 的影响。相同的设置和假设应用在图 9-6 中

9.4.2 宽共享总线架构

PicoServer 采用一个简单的宽共享总线架构以提供高内存带宽，并充分利用 3D 堆叠技术的优势。我们的总线架构是由在 M5 上的 SURGE 运行决定的。我们限制了对于 SURGE 的测试，因为它在电路基准的每个核中产生了有代表性的缓存未命中率。要探索总线架构的设计空间，首先要在单共享总线上模拟以 128 位 ~ 2048 位的不同位宽，并改变超缓存行大小以及以匹配总线宽度（从 16 ~ 256 个字节改变）。测量网络带宽（吞吐量的度量）以确定总线宽度对 PicoServer 的影响。如图 9-8a 所示，相对较宽的数据总线用来拓展网络性能，以满足较好的缓存未命中

需求。这是因为，共享总线上为窄总线带生成大量通信造成大量总线争夺，如图 9-8b，c 所示。当减少总线宽度时，总线流量的增加造成了延迟的超线性增加。减少总线的利用率意味着降低总线仲裁延迟，从而提高了网络带宽。宽总线宽度通过允许在一个事务中复制一大块数据，也有助于加快 NIC DMA 传输。为典型的 PicoServer 的 4、8 和 12 核配置一个 1024 位总线宽度是较为合理的。更多的核会导致网络性能饱和，除非采用更宽的总线。对于交错总线架构，却发现对于我们给出的一级缓存命中率，1024 位总线宽度足以处理总线请求。作为 16 个或更多核的架构和负载所产生的更高总线要求，通过提高一级缓存的未命中率——大于 10%，可使交错总线具有更高效率。一个交错总线架构增加出色总线请求的数量，这样可以寻址增加总线请求数。



a) 网络带宽与总线宽带　b) 总体总线负载延迟与总线宽度　c) 总线利用率与总线宽度

图 9-8　基于 SURGE 的各种共享总线架构，具有 16KB 的一级缓存（每个 I 和 D）的网络性能。假设这些实验的 CPU 时钟频率为 500MHz。随着处理器数量的增加，总线结构必须能够处理高带宽

9.4.3 片上 DRAM 架构

9.4.3.1 片上 DRAM 的作用

基于所述逻辑芯片面积估计，我们预计对于 4 核、8 核和 12 核的 PicoServers 的 DRAM 芯片尺寸分别为 $40mm^2$、$60mm^2$ 和 $80mm^2$。表 9-8 示出了对于 PicoServers 可供选择的片上存储器。例如，为了获得总大小为 256MB 的 DRAM，我们假定的 DRAM 是 4 层堆叠的。对于第 3 层的服务器，我们采用 8 层结构，因为它们在很大程度上依赖于系统内存的大小。对于目前 90nm 的技术，制造面积为 $80mm^2$，物理内存 256MB 的 4 层芯片堆叠是可行的。虽然今天（4 ~ 16GB）大内存量的物理内存在服务器群中是普遍的，在根据服务器工作负载并基于与数据中心专家讨论[41]的经验的基础上，我们相信服务器工作负载可以进行调整从而适合于具有更小的物理内存较小的系统。从对图 9-9 中所示的服务器应用程序的内存使用情况的测量，我们发现对于许多的服务器应用程序（除了 TPC-C 和 TPC-H），少量系统内存（约 64MB）被用户应用、数据和内核操作系统代码占据。该存储器的余下部分是

表 9-8　改变工艺技术对应的投影片上 DRAM 尺寸。面积估计是基于半导体源展望（Semiconductor SourceInsight）2005[45] 生成的。芯片 $80mm^2$ 尺寸类似于 90nm 的奔腾 M

	130nm	110nm	90nm	80nm
4 层每层 $40mm^2$ DRAM 堆叠	64MB	96MB	128MB	192MB
8 层每层 $40mm^2$ DRAM 堆叠	128MB	192MB	256MB	384MB
4 层每层 $60mm^2$ DRAM 堆叠	96MB	144MB	192MB	288MB
8 层每层 $60mm^2$ DRAM 堆叠	192MB	288MB	384MB	576MB
4 层每层 $80mm^2$ DRAM 堆叠	128MB	192MB	256MB	384MB
8 层每层 $40mm^2$ DRAM 堆叠	256MB	384MB	512MB	768MB

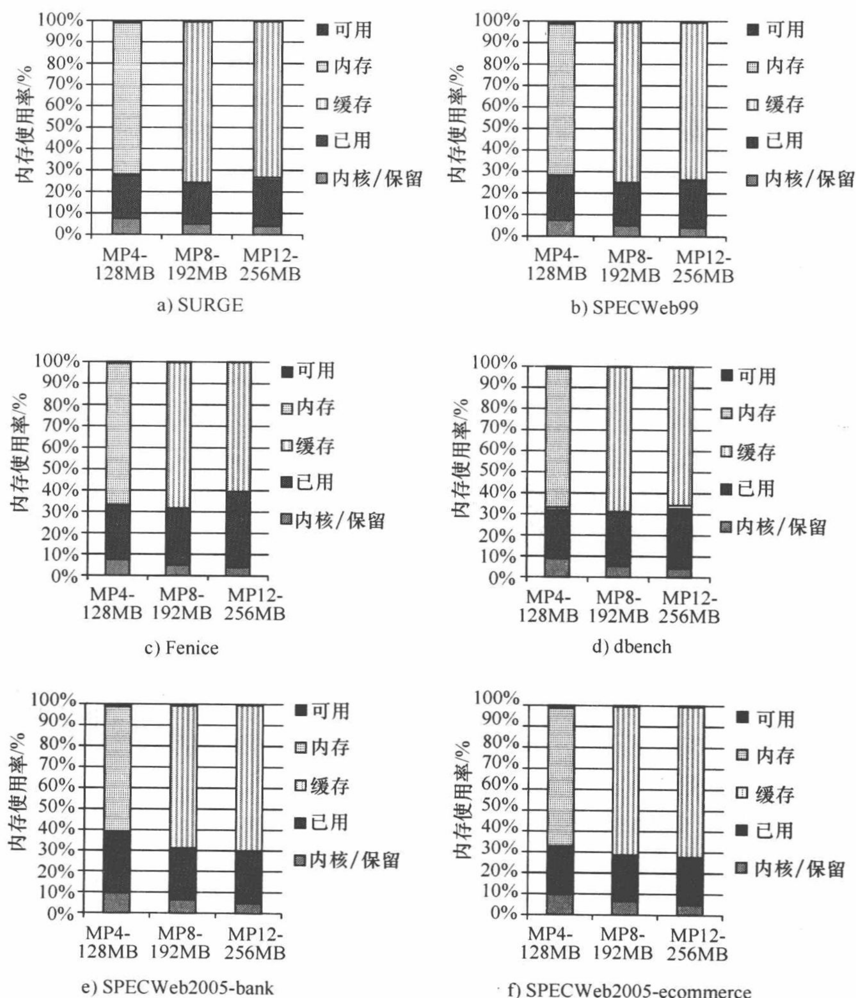
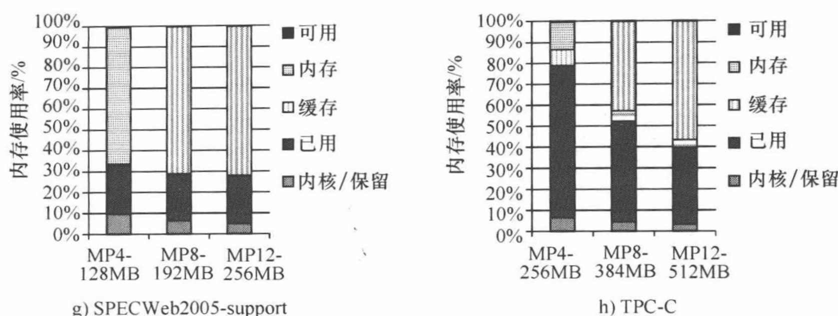


图 9-9　对于服务器基准测试的内存使用量（SURGE，SPECweb99，Fenice，dbench，SPECWeb2005，TPC-C）；TPC-H 被排除在外，因为它类似于 TPC-C 的内存使用情况



g) SPECWeb2005-support　　h) TPC-C

图 9-9　对于服务器基准测试的内存使用量（SURGE，SPECweb99，Fenice，dbench，SPECWeb2005，TPC-C）；TPC-H 被排除在外，因为它类似于 TPC-C 的内存使用情况（续）

空闲的或用作磁盘缓存。当我们考虑在 TPC-C 和 TPC-H 中将许多用户存储空间分配为用户级缓存时，对于 TPC-C 和 TPC-H 来说，这竟是真的。考虑到 256MB 可以集成在 4 层芯片上这一事实，片上的大部分 DRAM 可以被用作磁盘缓存。因此，对于需要小/中文件集的应用程序，一片上的 256MB 的 DRAM 足以处理客户机请求。

对于大型的文件集，有几种方案可供选择。首先，我们可以通过堆叠另外的 DRAM 芯片添加额外的片上 DRAM，如 8 层的情况。从表 9-2 的 ITRS 路线图可以看出，我们假设的堆叠芯片的数量是保守的。随着积极的芯片堆叠，我们可以添加更多的芯片堆叠来提高片上 DRAM 的产能，ITRS 预测在未来 2 ~4 年将超过 11 层。这是可能的，因为在逻辑层我们的功率密度是相当小的，即小于 $5W/cm^2$。另一种方法是增加一个作为磁盘缓存的次级系统存储器。对于这项研究，我们考虑其工作负载，发现这种辅助系统存储器的存取延迟可能慢至数百微秒，而不会影响吞吐量。访问延迟慢至数百微秒意味着消耗更少活动的或备用的电源闪存可作为次要系统存储器使用。这个想法一直在被探索[30,31]。因此，对于需要大量文件集的工作负载，我们可以建立具有快速的片上 DRAM 的不均匀的内存结构和相对较慢的片外辅助系统内存。快速的片上 DRAM 将主要包含代码、数据和一个小的磁盘缓存，而缓慢的系统内存将作为一个大的磁盘缓存设备。

9.4.3.2　片上 DRAM 接口

为了最大限度地提高的 3D 堆叠技术带来的好处，对于 PicoServer 的 3D 堆叠片上的 DRAM，常规的 DRAM 接口需要被修改。传统的 DDR2 DRAM 被设计的同时假设其具有少量引脚数，并使用地址复用和突发模式传输，以弥补有限数量的引脚。对于 3D 堆叠技术，没有必要使用熟悉的两相命令（RAS 然后 CAS）的窄接口和地址复用。代替的是，锁存和混合窄地址或数据的额外的逻辑需要可以被移除。请求地址可以作为一个单个的命令被发送而数据可以被大存储器中读出来。此外，传统的片外 DRAM，作为由多个 DDR2 DRAM 芯片组成的 DIMM 被提供。对于每个

请求，传统的片外 DIMM 接口接入多个 DDR2 DRAM 芯片。对于 3D 堆叠片上的 DRAM，每个请求只有一个子块需要被访问。结果是，3D 堆叠片上 DRAM 每个请求比外 DRAM 具有更低的功耗。图 9-10 示出了不用多路复用的读操作的例子。特别是，它表明的 RAS 和 CAS 地址请求被组合成一个单一的地址请求。DRAM 厂商已经提供了不需要地址复用的接口，如 Micron 的低延迟 DRAM[7] 和三星公司的 NetDRAM[55]。这表明叠层片上的 DRAM 所需的 3D 接口，仅需对现有的解决方案进行轻微的改变即可被实现。额外的芯片面积可以通过接口的简化得到，并可用于加快片上 DRAM 存取。通过牺牲更多的芯片面积到片上 DRAM 的子块，可以实现低至 10ns 的延迟㊀。

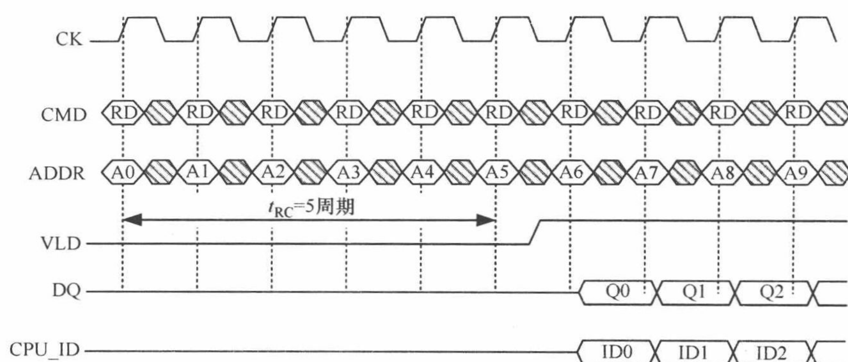


图 9-10　无地址复用的片上 DRAM 读时序图

9.4.3.3　片上 DRAM 刷新对吞吐量的影响

DRAM 的周期性要求每个 DRAM 单元被刷新。各 DRAM 单元的保持时间通常在工业标准温度被定义为 64ms，而在较热的环境中减小到 32ms。根据 9.4.5 节提出的热分析，在行业标准温度限制下，我们的最高结温是好的。因此，我们假设每个单元刷新周期为 64ms。然而，刷新电路通常由多个 DRAM 单元阵列间共享，以减少芯片面积开销，从而降低平均 DRAM 刷新间隔到约 7.8125μm，这大约需要 200ns 来完成。粗略地说，这意味着在每万 CPU 时钟周期中，一个 DRAM 块不能被访问持续几百个 CPU 时钟周期的时间。为了测量刷新周期的影响，我们模拟在 M5 中 DRAM 的刷新活动并观察 CPI 的开销。片上的 DRAM 的访问频率和所观察到的一级缓存未命中量直接相关。我们发现对于 5% 的一级缓存未命中率和在 500MHz 主频下的 12 核（PicoMP12-500MHz 运行 SURGE），将导致 0.03 CPI 刷新过载。这是因为许多一级缓存未命中与刷新指令不一致，使得其被执行时会导致只有最小的性能损失。

㊀ 在这个研究中，我们用一种保守的方法并且不考虑由于额外的次级块造成的延时减少。我们仅应用由于移除驱动片外信号而造成的延时优化。

9.4.4 一个 CMP 架构的多 NIC 需求

大型网络管服务器一个常见的问题是可以处理每秒到达几十万包的突发行为。中断聚结是处理这个问题的一种方法。它的工作原理是，当一个非关键事件发生时启动一个定时器。在计时器到期之前发生的任何其他非关键事件被合并为一个中断，以减少中断总数。即使有这样的技术，但是通过相对低频率的处理器接收中断的数量，如一个 PicoServer 核，也会超越它的限制。在仿真中，我们通过多个 NIC 解决这个困难，每个 NIC 对应于处理器的一个子集。对于具有一个 NIC 和片上的 DRAM 的 8 芯片多处理器架构，我们发现由于一个处理器不能由自身管理，使得每个处理器的平均利用率在 60% 以下。为了充分利用我们的多处理器架构中的每个处理器，每两个处理器插入一个 NIC。例如，具有 4 个处理器的结构将有两个网络接口控制器（NIC），一个 8 处理器架构将有 4 个网络接口控制器 NIC，依此类推。

尽管我们的模拟环境不支持它，但更理想的解决方案是使用一个更智能的单个的 NIC，可以按路线发送中断到多个 CPU，每个具有独立的 DMA 描述器和 TX/RX 队列。这可能是一个无论是具有多接口的 IP 地址还是利用智能方法负载平衡数据包到多个处理器的 NIC。这样的 NIC 需要在会话级跟踪网络协议状态。之前已经有 NIC 智能负载平衡的研究，以实现在平台上的最佳的吞吐量[21]。在更高层次的网络[46]，TCP 拼接和切换也是智能负载均衡很好的例子。

9.4.5 在三维堆叠中的热考虑

3D 堆叠技术的一个潜在的问题是热量包含的问题。为了解决这个问题，我们研究了三维堆叠上 PicoServer 架构的热影响。因为不能直接测量一个真正的 3D 堆叠平台上的温度，我们使用 Hotspot 3.1[28] 中的网格模型来建立 3D 堆叠模型。由于 FlowTherm 和 ANSYS 等有限元热仿真软件具有有限的 3D 堆叠工艺尺寸的资料，因此我们没有在研究中考虑。不过，热点的 RC 等效热流模型足以显示其趋势和 3D 堆叠潜在的问题。因为我们的工作旨在解决 3D 堆叠技术是否可在服务器领域展现其优势，而不是描述传热的细节，因此我们只提出传热的一般趋势。

从表 9-9 中可以看出，在 3D 堆叠技术中主要的热的容纳的贡献者是界面材料（SiO_2）和硅与空气之间的自由空气的界面。硅和金属能更有效传热。我们首先通过不同的条件配置 PicoServer 架构：①变化堆叠芯片的量；②变化堆叠中的主发热芯片块和逻辑芯片块的位置；③变化通常在堆叠芯片中间的 SiO_2 绝缘层的厚度。我们的基本结构是假设在 27℃ 的室温下直接连接到散热器的逻辑芯片。Hotspot 需要材料和功率密度信息来计算稳态温度。因此我们从本章参考文献［37，44，57］提取三维堆叠特性，进而基于面积和功率为每个组件在组件级别分配功率密度。平台级处理器、外设、全局总线互连等各组分被建模。生成的 PicoServer 架构下的最大结温如图 9-11 所示。

表 9-9　对于常见材料的硅器件热参数

	热导率/(W/m·K)	热沉/(J/m³·K)
Si	148	1.75×10^6
SiO_2	1.36	1.86×10^6
Cu	385	3.86×10^6
25℃的空气	0.026	1.2×10^3

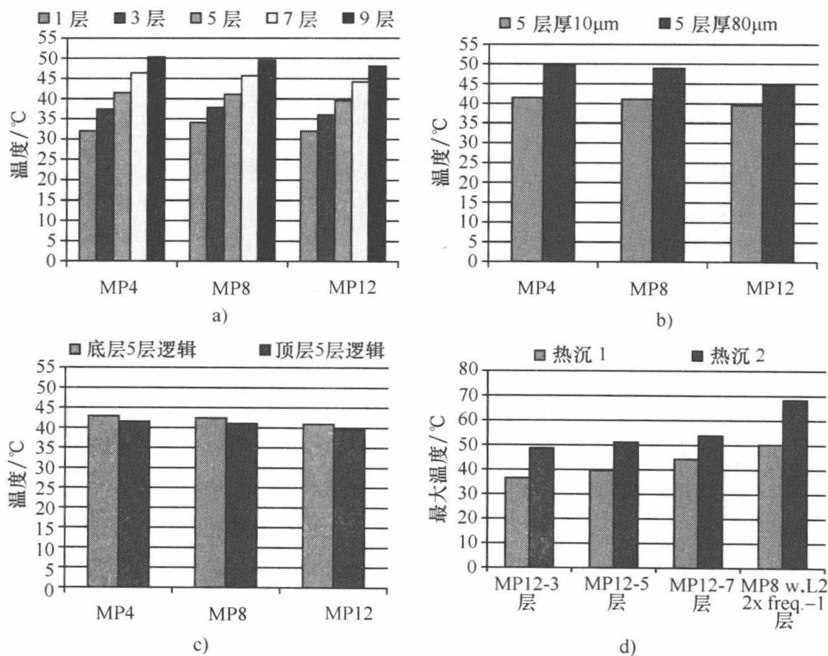


图 9-11　Hotspot 灵敏的实验中最大结点温度：a）不同层的数目；b）不同的 3D 界面的厚度；c）改变逻辑芯片的位置；d）最大结温对散热器质量分析。假设在 500MHz 的时钟频率下计算功率密度。我们改变基于叠层数量的片上存储器大小。且假设层上无片上存储器

图 9-11a 显示了堆叠层数目的灵敏度。我们发现每增加一层，最高结温增加 2～3℃。有趣的是，当我们增加芯片面积，最大结点温度降低。我们认为，这是由于我们的布局规划和封装的假设。这需要进一步分析，有待我们将来研究。图 9-11b 显示出三维堆叠电介质界面的敏感性。我们比较了 SiO_2 厚度（界面材料）为 10μm 和 80μm 的情况。在本章参考文献［17，37，44，57］我们发现对于 3D 堆叠界面材料的最大厚度不超过 10μm。选择 80μm 的点，以显示随着厚度大幅增加热容的影响。这导致结温增加 6℃。而值得注意的是，材料厚度的改变没有带来显著的变化。由于片上总线的频率相对较慢，我们假定电介质界面厚度增加没有增加总线延时时间。图 9-11c 显示了堆栈中位置的灵敏度——顶层位置或底层位置。我们发现主发热芯片对散热器的位置不敏感。

我们还对散热片质量的影响进行分析。通过改变散热器的配置，模拟了一个高成本散热器（散热器 1）和一个低成本散热器（散热器 2）。图 9-11d 显示了 3D 堆叠技术对散热器质量的影响。它清楚地表明，一个低成本的散热器可以在三维堆叠技术平台上使用。上述结果表明，热容纳是 PicoServer 架构的主要限制。由于功率密度相对较低，不超过 $5W/cm^2$，因此，最大结温度不超过 50℃。三维通孔也可以作为热通道，这是我们在分析中没有考虑到的，然而其改善情况是可以预期的。一个智能布局将会分配发热层（处理器层）和热层相邻，使得大多数的热被转移到散热器。本章参考文献［19，25］的研究分别支持了我们的结论。

9.4.6　将闪存集成到 PicoServer 的影响

本节考察直接连接 NAND 闪存器件到 PicoServer 架构的结构的影响，同时它也是对集成到 PicoServer 3D 堆叠闪存器件情况的研究。由于闪存密度的迅速提高，闪存正在成为一个有吸引力的集成到服务器平台的存储设备。闪存在服务器领域有两个主要的使用模式，即固态盘（SSD）和存储装置。人们普遍认为，闪存集成提高了整体服务器的吞吐量，而使系统内存和磁盘驱动的功率更低。例如，当闪存用作存储装置并被分配作为磁盘缓存时，较高密度 Flash 允许我们实现较高的缓存命中率，这比 DRAM 的功耗更低。较高的缓存命中率降低存取磁盘，这使得系统性能改进和磁盘功率得到减少。

然而，集成闪存到服务器平台并不简单。成功整合闪存到一个服务器平台有两大挑战。①提高传输延迟到 Flash；②提供足够的内存（RAM），以有效地管理闪存。与典型的存储器装置相比较，例如 DRAM，NAND 闪存显示较高的总访问延迟（见表 9-10），这主要由于高传输延时（低带宽窄 8 位或 16 位的接口），用于典型架子外的闪存设备。虽然对于 SLC NAND 闪存页面，到内部缓冲区的读取延迟大约是 25μs，但对于 NAND 闪存芯片读取几个 KB 的传输延迟来说是相当高的。减少传输延迟的方法之一是利用 3D 堆叠。它能够使用更宽的总线每周期访问更大量的数据，从而减少传输延迟。为了更多的节约能源，可以减少等待时间允许关键数据从 DRAM 移到闪存。

表 9-10　ITRS 2007 年后路线图中的内存发展趋势

	2007	2009	2011	2013	2015
NAND Flash-SLC①/(μm^2/bit)	0.0130	0.0081	0.0052	0.0031	0.0021
NAND Flash-MLC①/(μm^2/bit)	0.0065	0.0041	0.0013	0.0008	0.0005
DRAM 单元浓度/(μm^2/bit)	0.0324	0.0153	0.0096	0.0061	0.0038
Flash 写/擦周期——SLC/MLC②	1E+05/ 1E+04	1E+05/ 1E+04	1E+06/ 1E+04	1E+06/ 1E+04	1E+06/ 1E+04
闪存数据保持/年	10~20	10~20	10~20	20	20

① SLC——单级单元，MLC——多级单元。

② 从先前估计的工作中写/擦除周期 MLC 闪存[34]。

此外，被需要来有效管理 NAND 闪存子系统的内存（RAM）容量与其容量成比例。虽然 NAND 闪存仍然可以用少量的存储器管理，但这些类型的子系统只显示有限的读/写带宽并加速了闪存损耗。为了满足存储器（RAM）提供高带宽、高寿命和高容量的 NAND 闪存子系统的要求，DRAM 通常集成到存储的闪存管理代码和数据的 NAND 闪存储器子系统。然而，实现在 NAND 闪存子系统所需要专用 DRAM 装置的成本是可估计的，低效的。通过巩固整个服务器平台的代码和数据到一个单一的 DRAM 模块可节省成本和提高效率。3D 堆叠系统集成的益处在于允许闪存管理代码（见图 9-12）和数据驻留在与系统中的其他部件共用的 DRAM 中。

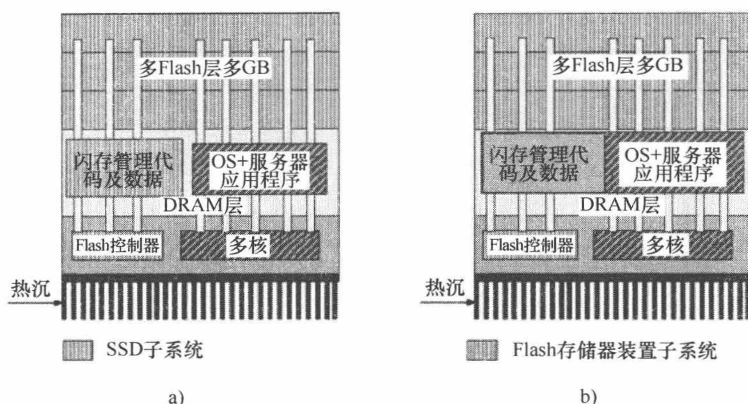


图 9-12 三维堆叠闪存架构用于：a）固态使用模型；b）存储器装置的使用模式

对于固态盘（SSD）使用模式，3D 堆叠可以用作实现节能的片上接口的方式，这种接口可以取代传统的硬盘驱动器接口。图 9-12a 显示了这种方法。使用与 CPU 核相同的工艺技术，将 SSD 控制器逻辑放置在芯片上。闪存管理的低层细节（错误检查、损耗平衡和缓冲管理）被从处理器中隔离，这样提供了简单的接口。对于片上集成 DRAM 来说，则很容易满足其闪存管理的内存要求。一个三维堆叠的 SSD 可以提供：①较低的功率；②较高的随机访问吞吐量；③更低的延迟；④比硬盘更好物理稳健性。固态硬盘除了包括闪存，还包括一个控制器和缓存 RAM。

在一个采用 3D 堆叠技术的 PicoServer 中，硬件中留有一个固态硬盘接口的可以让驱动器无须修改即可进行工作。外部 SSD 的其他好处包括具有更高带宽的接口，具有直接进入主存储器或直接进入处理器缓存的数据传送选项，并具有允许 PicoServer 核使用应用专用算法来控制闪存的能力。由于各装置[35]的不同特征，使用 SSD 和常规的基于磁盘的存储装置组合，可以提供更高的性能。对高带宽、高密度的存储来说，硬盘是最有效的，同时 Flash 提供了更低的延迟（尤其是当读的时候）和更高的 IOP（每秒读写数）。为整体的性能提高，利用算法动态地放置读控制数据在 SSD 中，同时写控制的数据被移动到 HDD。

对于存储设备的使用模式，已经出现了若干建议，这些建议涉及广泛应用的，

包括磁盘缓冲区[33]、磁盘缓存[29,30,31]和代码/数据存储[42,49]。图 9-12b 显示了其物理结构是为何与三维堆叠的 SSD 的物理结构十分相似，除了被提供给处理器核心的闪存之间低级别的数据传输控制外。这种方法增加了软件的复杂性，但是实现了比固态硬盘（SSD）使用模式更高的带宽和更有效的管理。这是因为 Flash 管理是由比闪存控制器具有更多的计算能力的处理器执行。当闪存用作磁盘缓冲时，它被用作 DRAM 和磁盘之间的暂存缓冲器。当数据从 Flash 缓冲器读出时，我们可以通过关闭磁盘获得更长的时间来实现节能。该方案还能减少系统中 DRAM 的数量，并节省系统能量的 30%～40%。同样的，通过扩展基于 DRAM 的磁盘缓存的标准，闪存磁盘缓存在操作系统是很容易实现的。数据集中型服务器应用程序，如 Web 服务器，需要大量的磁盘缓存。通过使用 Flash 取代部分 DRAM，可以降低空闲功耗，可以有更多的总缓存容量（由于较高的闪存密度），使得性能得到提升，从而节约能源。三维堆叠技术通过更快速地满足缓存数据的请求，而增强了该方案。缓存的网页从第二 Flash 存储迁移到 DRAM 的主要页面缓存，这样宽的 3D 堆叠接口连接可以更快的、用更低的总线能量完成这些数据传输。存储设备的使用模式还有其他的益处。使用小型增强型的微架构 Flash 控制器，通常从 NAND 闪存被加载并驻留在 DRAM 中的代码可能直接驻留在 NAND 中。在本章参考文献［42］中描述的下一个码块预测技术和在本章参考文献［49］中概述的“基于需求页”的体系结构已经表明其潜在益处。通过降低能源成本和延迟，同时交换在 DRAM 和 Flash 之间的页面，三维堆叠技术可以帮助这个方案实施。对于相同的 DRAM 和闪存容量，总能量将进一步降低，因为程序执行得更迅速，并且使得在 DRAM 中和其他系统部件，如供电电源中，消耗较少的空闲能量。

根据本章参考文献［29，30，31］的研究结果，我们提出三维堆叠 Flash 作为存储器设备集成到一个 PicoServer 架构的例子。

我们首先进行运行的服务器工作负载操作系统管理磁盘缓存行为的分析。图 9-13 显示在 Web 服务器系统存储器的磁盘缓存访问行为。这表明，服务器工作负载访问文件的行为是一种短尾分布行为，即其中磁盘缓存的很大一部分不经常访问，只小部分磁盘缓存被频繁访问。在访问磁盘缓存中不经常访问的文件时，将导致变化的访问延迟。图 9-14 显示了这种情况下的服务器吞吐量。对于不同的访问延迟，我们可以观察到吞吐量常数在十到几百微秒之间。这是因为我们可以通过服务器工作负载的自然多线程性隐藏不经常访问的磁盘缓存文件访问延迟的时长。利用这一特性的一种方法是闪存集成为第二级磁盘缓存，而代替磁盘分配缓存中的大量的 DRAM 分配。

因为 NAND 闪存比 DRAM 消耗少得多的功率并具有比 DRAM 大 4 倍的密度（见表 9-10 和表 9-11 所示），在服务器存储器系统中集成闪存被认为更节能且有更大的系统存储容量。更大的磁盘缓存可以减少磁盘缓存的未命中率，让硬盘停转时间更长。如我们在表 9-11 中所示，在空闲模式下硬盘消耗了一定的电量。

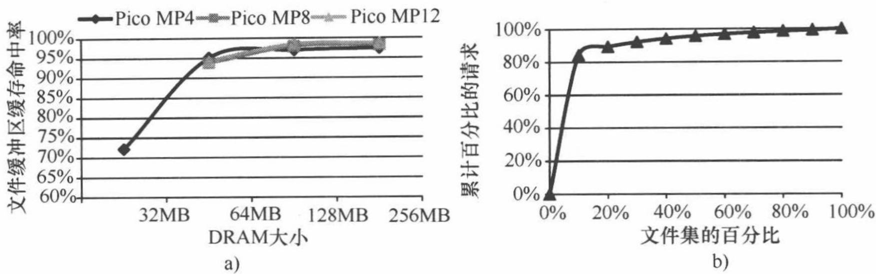


图9-13 a）在服务器端对于客户端请求的磁盘高速缓冲存储器存取行为，分别对于4核、8核、12核PicoServer配置和不同的DRAM大小；b）客户端请求行为的典型的累积分布函数。请求的90%对应于20%的Web内容文件

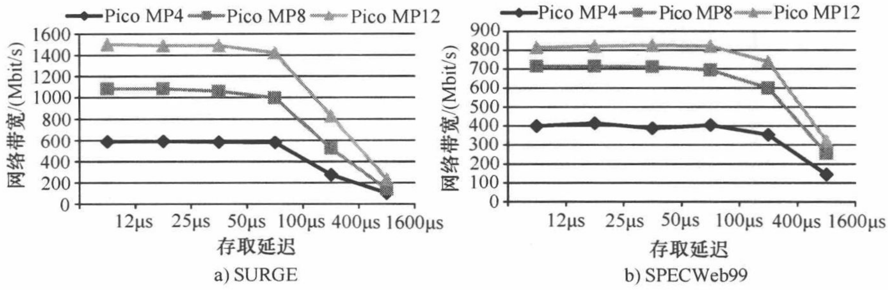


图9-14 变化到达二级磁盘缓存时的访问延迟，测量的全系统仿真网络带宽。假设一个有1GB更慢速存储器的128MB的DRAM，分别测得配置为4核、8核、12核PicoServer 500MHz的带宽。二级磁盘缓存可以容忍几百微秒的访问延迟，同时提供平等的网络带宽

表9-11 DRAM、NAND-basedSLC/MLC闪存和硬盘的性能、功耗和成本

	动态功耗	待机功耗	读延迟	写延迟	擦延迟
1Gb DDR2 DRAM	878mW	80mW①	55ns	55ns	N/A
1Gb NAND-SLC	27mW	6μW	25μs	200μs	1.5ms
4Gb NAND-MLC	N/A	N/A	50μs	680μs	3.3ms
HDD②	13.0W	9.3W	8.5ms	9.5ms	N/A

① DRAM待机功率处于活动模式。待机功耗在省电模式下为18mW。

② 750GB硬盘驱动器的数据[8]。

集成闪存到服务器所取得的好处也可以被应用到PicoServer中。图9-15和图9-16显示了使用3D堆叠技术集成闪存到PicoServer的两种配置。图9-15a显示了第一种配置，在图9-16a中将闪存作为分立元件集成。采用3D堆叠技术堆叠8层闪存层产生一个分离的Flash芯片—分立元件。这种闪存芯片通过PCB板布线连接到PicoServer。一个能够每秒提供数百兆字节的传输的片外I/O接口（通常是

PCI Express 高速）应用在 PicoServer 和分立的闪存芯片之间。闪存通过在有序 PicoServer 内核上运行的操作系统管理（耗损均衡管理和 Flash 命令接口管理）。分立的闪存芯片的存储容量不受 PicoServer 芯片面积的限制，而且由于 Flash 的功耗更低，允许我们堆叠比 PicoServer 更多的芯片。因此，数十 GB 分立的闪存芯片可以集成到 PicoServer 架构中。具有大型的文件集和适度的 I/O 带宽需求的服务器负载将最大的受益于这种配置。这样的分立闪存芯片配置潜在缺陷是 PCIE 接口的空闲功耗。在 PCIE I/O 引脚处，即使当其不活跃时，空闲电源管理仍必须进行。

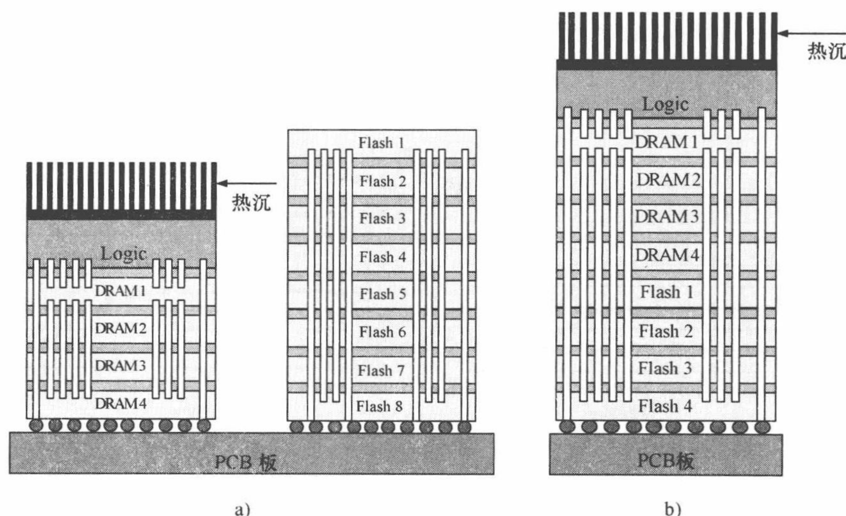


图 9-15 a) 8 层堆叠的 Flash 芯片作为分立的部件（通过 PCB 布局）集成到 PicoServer 中；b) 4 层堆叠的 Flash 芯片直接集成到 PicoServer 中

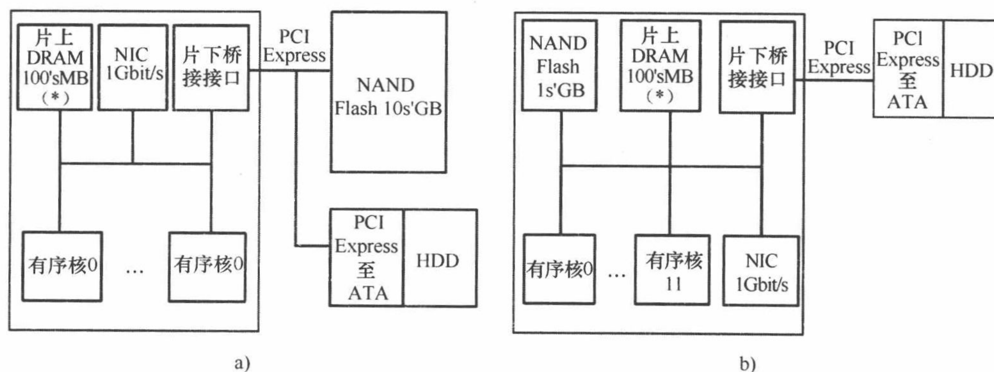


图 9-16 a) 8 层堆叠的 Flash 芯片作为分立的部件（通过 PCB 布局）集成到 PicoServer 中的高级框图；b) 4 层堆叠的 Flash 芯片直接集成到 PicoServer 中的高级框图

图 9-15b 和图 9-16b 所示的第二配置直接集成 Flash 到 PicoServer（在前面的章节中讨论）。它使用三维堆叠技术在 DRAM 芯片上直接堆叠 4 层闪存芯片。闪存芯片通过宽的共享总线连接到 PicoServer 中的其他组件。宽片上共享总线接口具有每秒数十倍的千兆字节带宽。Flash 还可以成功进行损耗均衡管理和闪存指令接口管理，这是在有序 PicoServer 内核中运行操作系统实现的。闪存容量受在 PicoServer 片上的 DRAM 和逻辑元件的芯片面积限制。因此，闪存容量预计有几千兆字节的大小。预计这种需要小文件集和高 I/O 带宽的服务器工作负载将受益于这种配置。

9.5 结果

为了评估 PicoServer 架构，两个重要的指标是吞吐量和功耗。吞吐量可被测量为网络带宽或每秒传送量。由于它是一种多少个请求被服务的量度，因此是衡量整个系统的性能的良好指标。在本节中，我们首先将不同配置的 PicoServer 和其他架构在吞吐量方面进行对比，接着对功率方面进行对比。由于 PicoServer 尚未实现，我们使用分析模型和发表的数据结合的方式对关于各种部件的功率消耗进行保守估计。最后，提出了一种显示 PicoServer 架构能源效率的方式——帕累托图（pareto）。

9.5.1 整体表现

图 9-17 和图 9-18 显示出一些 1 ~3 层负载运行的吞吐量。每栏在 3 个方面显示其对吞吐量的贡献：①没有二级缓存和一个窄（64 位）总线的基准；②有二级缓存基准；③没有二级缓存和 3D DRAM 堆叠的宽总线基准。因此，我们可以比较有无二级缓存的 3D 堆叠的技术的影响。图 9-17 显示了单独使用 3D 堆叠技术提高的整体性能等价于或超过使用二级缓存。一个固定核数的公平比较，例如，可能是将一个 Pico MP4-1000MHz 与传统的没有 3D-1000MHz 的 MP4 CMP 的比较。一般情况下，产生适度的高缓存未命中率（SURGE、SPECweb99、SPECweb2005 和 dbench）的工作负载，显示了采用 3D 堆叠技术可以使其得到显著改善。因为涉及较低的缓存未命中率的视频流的计算，Fenice 没有得到显著的改善。有趣的是，第二层基准电路（SPECweb2005 的脚本语言），和已经针对单线程性能而设计的 OO4 配置相比表现良好。

对于 OO4 配置，由于在单处理器二级缓存的延迟可能比到一个大容量 DRAM 的存取延迟小，使得对于仅具有高带宽 3D 堆叠的片上 DRAM 不太具有吸引力，因此我们结合具有二级缓存和 3D 堆叠的影响。我们发现对于 OO4 配置，3D 堆叠提高了其 15% 的性能。当我们比较一个采用无 3D 堆叠的 PicoServer 架构的 OO4 结构时，一个操作在 500MHz 下的 PicoServer MP8 表现比用具有小的 16KB 和 256KB 的一级和二级缓存的 4GHz OO4 的处理器更好。对于类似的芯片面积比较，由于对于

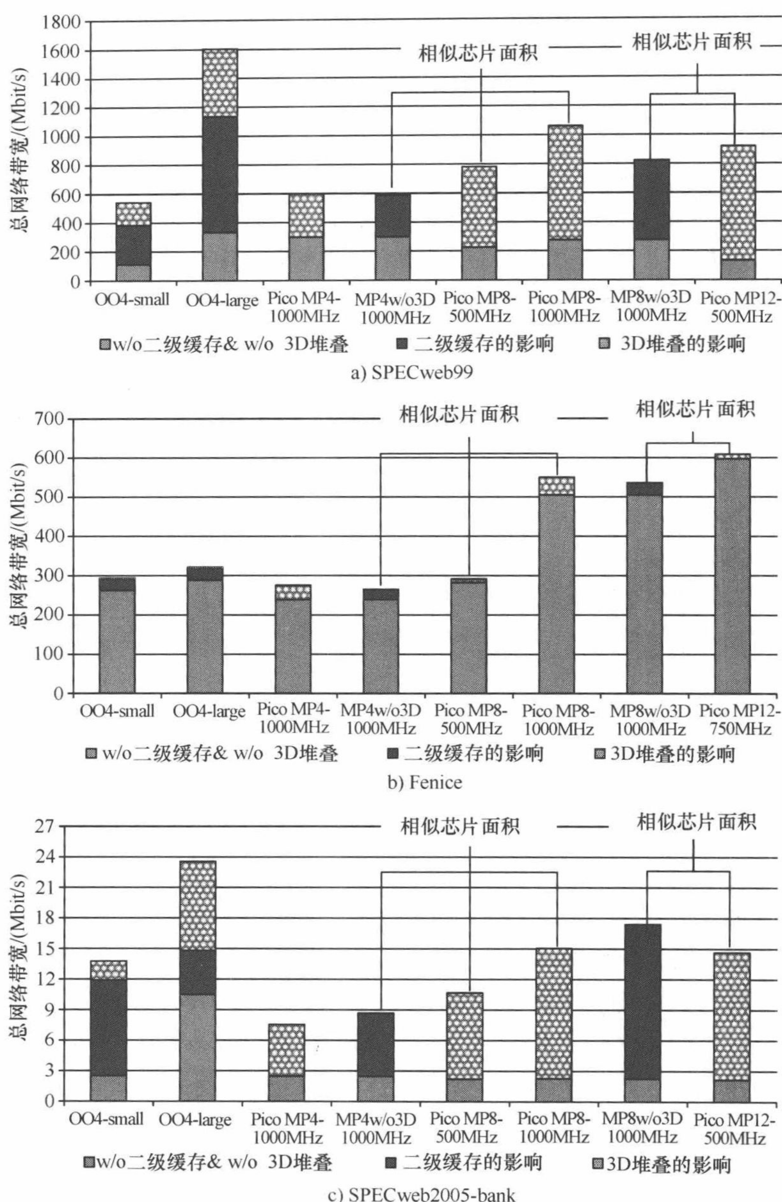


图9-17 变化处理器频率和处理器类型时所测量的吞吐量。对于PicoServer CMP我们固定的芯片上的数据总线宽度为1024位，总线频率250MHz。对于一个类似奔腾4的配置，我们把NIC放到PCI总线上并假定存储器总线频率为400MHz。对于一个没有采用3D堆叠技术的MP4、MP8配置，为了公平起见，我们假设不支持多线程和2MB大小的二级缓存。假设外部存储器总线频率为250MHz（SPECweb99，Fenice，SPECweb2005-bank块）

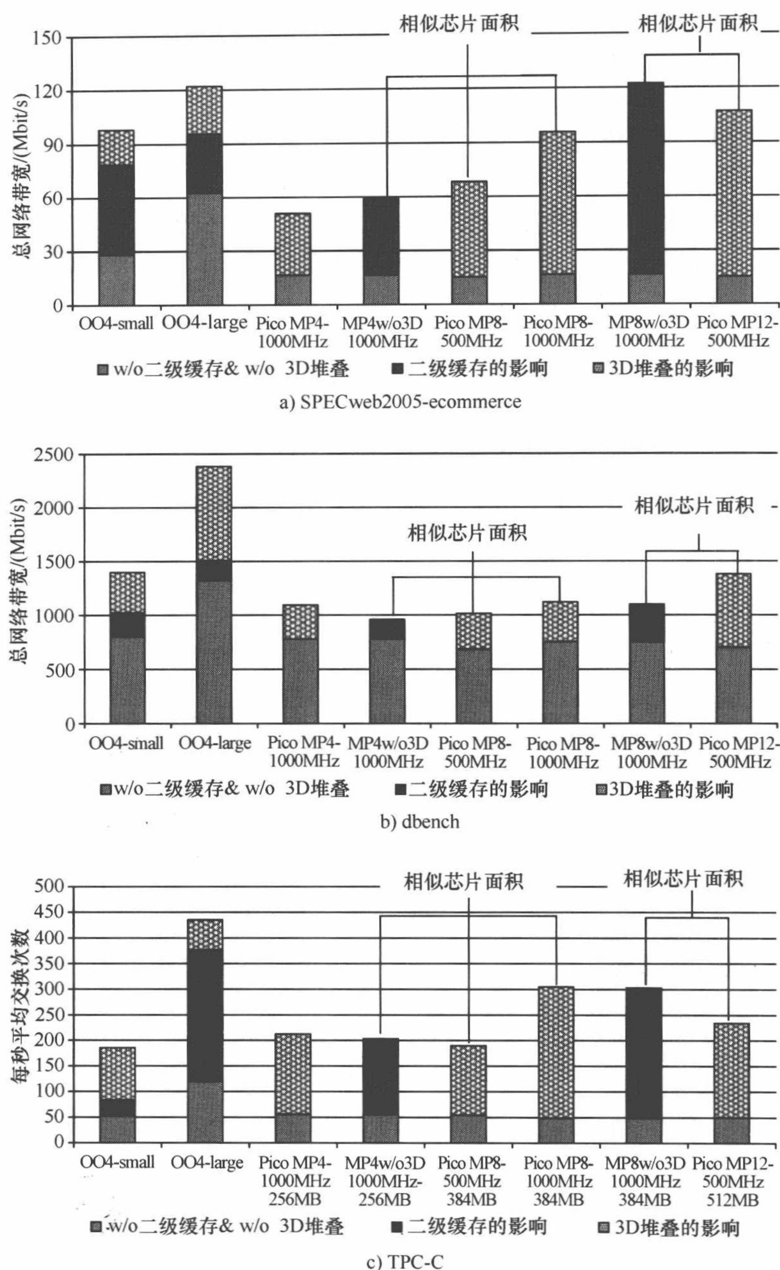


图 9-18　改变处理器频率和处理器类型（SPECweb2005-ecommerce，dbench，TPC-C）时所测量的吞吐量。与我们在图 9-17 中使用相同的假设

128KB 的一级缓存和 2MB 的二级缓存来说，大型 OO4 需要额外的芯片面积，因此我们认为 PicoServer MP8 和小的 OO4 架构的比较是公平的比较。

如果我们假设在传统的没有采用3D 堆叠技术的 CMP MP4/8 中二级缓存所占据的区域被替换为另外的处理核——这是使用三维堆叠技术带来的一个好处，对于类似面积的芯片吞吐量的比较，可以基于以下系统进行：①一个 Pico MP8-500MHz 的 MP4 和没有 3D-1000MHz 的常规的 MP4 比较；②一 Pico MP12-500MHz 的 MP8 和没有 3D-1000MHz 的常规的 MP8 比较（对于 Fenice 和一个 Pico MP12-750MHz）。研究结果表明，平均而言，额外的处理元件，加上减半内核时钟频率可以提高吞吐量并显著节省电源，见 9.5.2 节。然而对于商务计算型受限负载像 Fenice、SPECWeb2005-银行和 SPECWeb2005-电子商务，Pico MP12-500MHz 并不比一个没有 3D-1000MHz 的常规的 MP8 表现得更好。对于 SPECWeb2005-银行和电子商务，引进一个 2MB 大小的二级缓存极大减少了缓存未命中率，降低了由于增加更多的内核带来的好处，同时降低了内核时钟频率数量。由于 I/O 调度，对 TPC-C、PicoMP12-500MHz 也表现得不好。然而，我们预期对带有 TPC-C 优化的 I/O 调度算法的操作系统内核，Pico MP12-500MHz 能表现得更好。我们用于添加额外的核心面积估计相当保守，同时我们建议添加多个核，使吞吐量有进一步的改善。

9.5.2 总体功率

处理器的功耗仍在 PicoServer 架构整体功耗中占主导地位。图 9-19 显示了对于服务器应用程序运行且基于我们的功耗估计技术的平均功耗。我们发现 500MHz 核心时钟频率的 PicoServer，在 90nm 工艺技术下，估计要消耗 2 ~ 3W。大多的总功耗是由简单的系列循环处理器消耗。当处理器数目增加时，网络接口控制器也增加，因此网络接口控制器（NIC）的功率也是总功率的主要部分。然而，如第 9.4.4 节所述，为此架构设计的一个智能网络接口控制器（NIC），可以作为一种更先进的、更有效率的制造，且将只需要一个。由于三维堆叠造成的 DRAM 功率可观的减少量已被观察到，则这个简化的片上 DRAM 接口需更少的 DRAM 子块同时访问每个请求。其他部件在处于适中的访问率和翻转率的情况下，其在整体系统的功耗中仅占一小部分。

将 PicoServer 架构与其他架构比较，可以看到在芯片面积相同的情况下，Pico MP8/12-500MHz 与常规的没有采用 3D 堆叠技术的具有 1000MHz 的二级缓存的 MP4/8 比较时，仅用了少于一半的功率。我们也回顾了第 9.5.1 节对于相同的芯片面积下的性能考虑，PicoServer 架构的性能优于常规的 CMP 配置平均 10% ~ 20%。此外，正如在上一节中表明，我们能使用少于奔腾 4 处理器功率的 10% 来执行同等任务。在 90nm 技术时，可以预计对于典型 PicoServer 平台的功率是满足 ITRS 注明的功率约束的。这表明了实现超小外形平台服务器应用类型的可能。

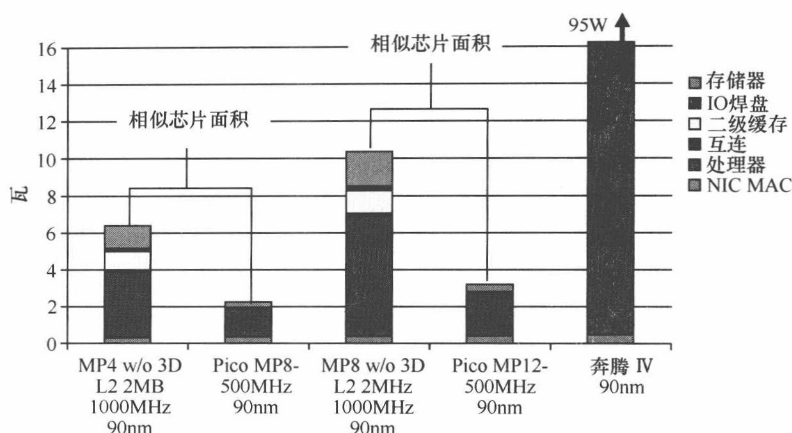
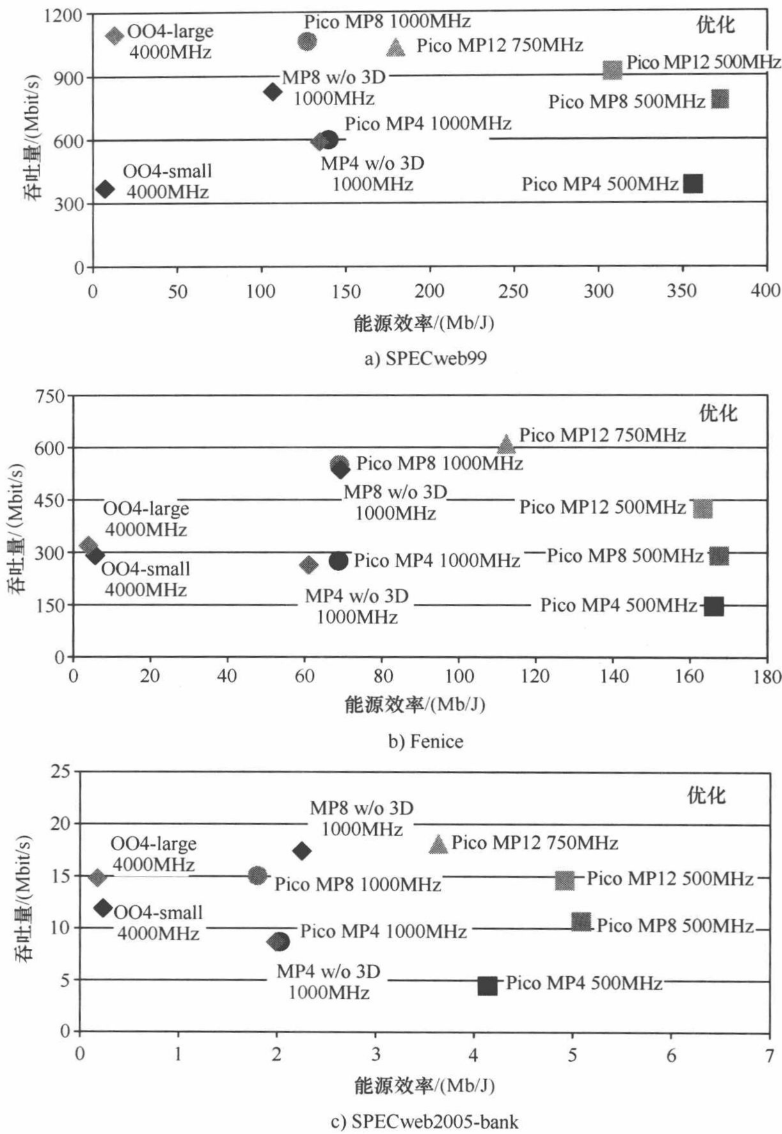


图 9-19　采用 90nm 工艺下的 3D 堆叠技术的 4 核、8 核、12 核处理器 PicoServer 架构的平均击穿功率。由于内核功耗是功耗的主要部分，每个工作负载的估计功率没有改变。我们预计在 90nm 的工艺下有 2～3W 的功耗。在 1GHz 频率下，没有采用 3D 堆叠技术的 MP8 估计要消耗 8W 的功耗

9.5.3　能源效率的帕累托（Pareto）图

在图 9-20 和图 9-21 中，对于 PicoServer 我们提出了一个帕累托（Pareto）图表来描绘能源效率（Mbs 每焦耳）和吞吐量（我们只列出了主要的工作负载）。在此图中的点示出了大量乱序处理器，分别是常规的没有采用 3D 堆叠技术的 CMP MP4/8 的处理器和有 4 个、8 个与 12 个核心的 PicoServer。y 轴代表 Mbit/s 级的传输，在 x 轴代表 Mb/J 和每焦耳的传输。从图 9-20 和图 9-21 可以看出，对于给定的能源效率/吞吐量约束的处理器数量和频率，我们是可能得到最优配置的。

此外，从图 9-20 和图 9-21 中我们发现，与没有 3D 堆叠技术传统的芯片多处理器架构相比，PicoServer 架构在适中的 500MHz 核心频率下具有 2～4 倍能源效率。主要的功率节省归因于可以使核心时钟频率减少的同时提供高吞吐量的三维堆叠技术。对于我们绘制的数据点，在 PicoServer 架构中，当比较 PicoMP4-500MHz、MP8-500MHz 和 MP12-500MHz 时，系统级能效的最优点可以被确定。在吞吐量达到并行之前的热点及边际递减极限到达之前，此处的吞吐量极限是通过增加处理器而增加的并行处理。并行处理的增加引起了关于到低效率的中断平衡、内核进程/线程调度和资源分配的问题，这些问题会导致减少返回。独立研究已经表明，OS（操作系统）可以被调谐扩展到多个核。本章参考文献［18］和［54］就是这样的例子。然而，我们觉得这项工作需要进一步的调查，将在以后研究。



a) SPECweb99

b) Fenice

c) SPECweb2005-bank

图 9-20 在 90nm 工艺技术下的能源效率和性能的帕累托（Pareto）图。三维堆叠技术使新型的节能 CMP 架构成为可能（SPECweb99，Fenice，SPECweb2005 块）

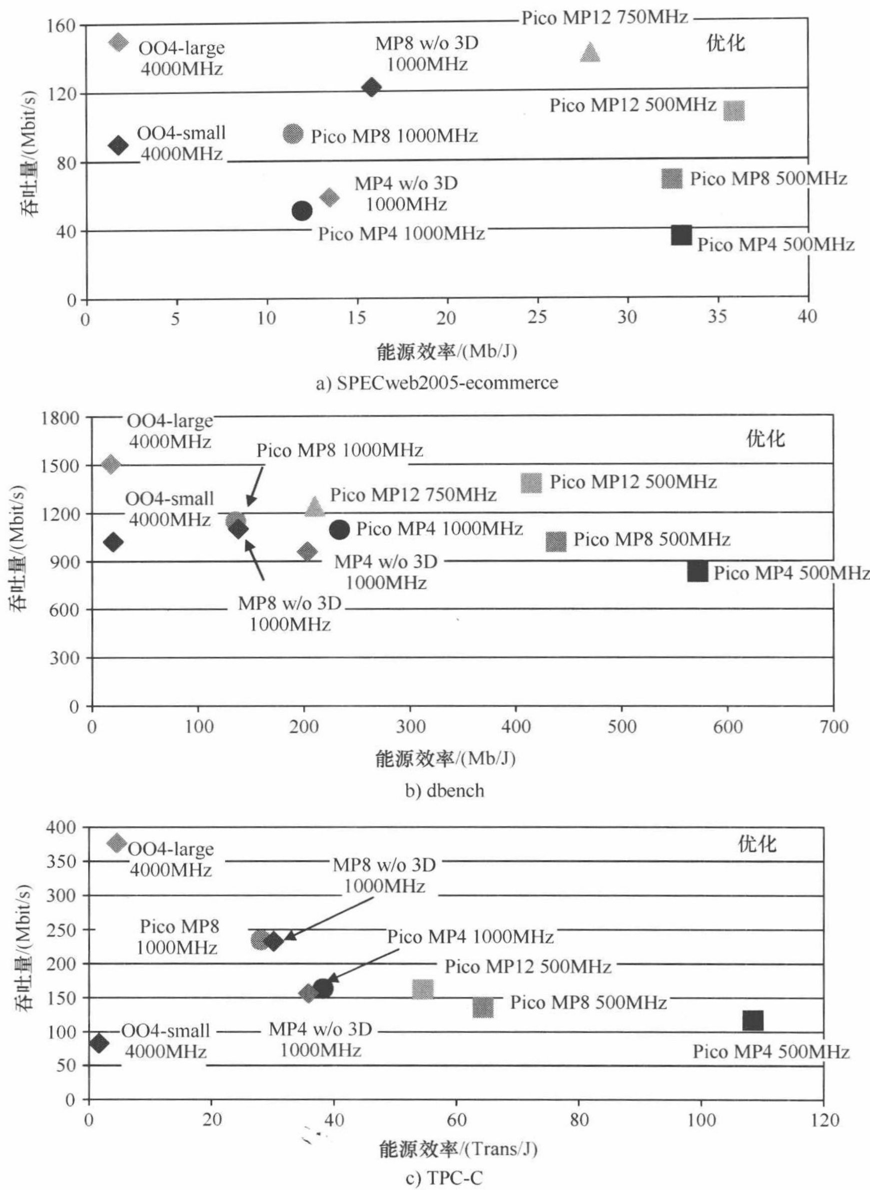


图 9-21 在 90nm 工艺技术下的能源效率和性能的帕累托（Pareto）图。三维堆叠技术使新型的节能 CMP 架构成为可能（SPECweb2005-ecommerce，dbench，TPC-C）

9.6 结论

在本章中，我们说明了3D堆叠技术可以被利用来构建高效节能的服务器。对于广泛的服务器工作负载，由此产生的系统在一个紧凑的外形尺寸下有显著的能源效率。90nm工艺下，一个500MHz下的12路PicoServer，预计可以在3W的功耗内提供1Gbit/s的网络带宽。这些功耗值是没有采用3D堆叠技术的多核架构的功耗值的两到三倍，并且优于使用一般用途处理器一个量级。相比传统的有2MB二级缓存的8路1GHz的多处理器芯片，一个在500MHz下运行的具有等效面积的12路PicoServer，能使得能量效率有超过2倍的改进。预计绝对功率值会随工艺技术变化。对于基于PicoServer平台的65nm和45nm工艺技术节点来说，我们希望看到更多的内核和更低的功耗。

通过广泛和低延迟的互连，紧密结合大量内存到内核周围，还带来了额外的收益，降低了系统复杂度，并用不一致的访问延时来创造已实施系统内存的机会。三维技术使高吞吐量、低功耗的DRAM核接口成为可能。由于片上的DRAM访问延迟和二级缓存的访问延迟是可比较的，则二级缓存芯片面积可以被额外的核替换，使得核心时钟频率降低，同时实现更高的吞吐量。

致　　谢

这项工作是由美国国家科学基金会，Intel公司和ARM公司的支持。

参考文献

1. ARM 11 MPcore. http://www.arm.com/products/CPUs/ARM11MPCoreMultiprocessor.html.
2. FaStack 3D RISC super-8051 microcontroller. http://www.tachyonsemi.com/OtherICs/datasheets/TSCR8051Lx_1_5Web.pdf.
3. The Micron system-power calculator. http://www.micron.com/products/dram/syscalc.html.
4. National semiconductor DP83820 10 / 100 / 1000 Mb/s PCI ethernet network interface controller.
5. OSDL DataBase Test Suite. http://www.osdl.net/lab_activities/kernel_testing/osdl_database_test_suite/.
6. $(LS)^3$-libre streaming, libre software, libre standards an open multimedia streaming project. http://streaming.polito.it/.
7. RLDRAM memory. http://www.micron.com/products/dram/rldram/.
8. Seagate Barracuda. http://www.seagate.com/products/personal/index.html.
9. SPECweb2005 benchmark. http://www.spec.org/web2005/.
10. SPECweb99 benchmark. http://www.spec.org/osg/web99/.
11. Sun Fire T2000 Server Power Calculator. http://www.sun.com/servers/coolthreads/t2000/calc/index.jsp.
12. ITRS roadmap. Technical report, 2005.

13. K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat. 3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. *Proceedings of the IEEE*, 89(5):602–633, May 2001.
14. P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *Measurement and Modeling of Computer Systems*, pp. 151–160, 1998.
15. N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt. The M5 simulator: Modeling networked systems. *IEEE Micro*, 26(4):52–60, Jul/Aug 2006.
16. B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. P. Shen, and C. Webb. Die stacking (3D) microarchitecture. In *International Symposium on Microarchitecture*, December 2006.
17. B. Black, D. Nelson, C. Webb, and N. Samra. 3D processing technology and its impact on iA32 microprocessors. In *Proceedings of International Conference on Computer Design*, pp. 316–318, 2004.
18. R. Bryant, J. Hawkes, J. Steiner, J. Barnes, and J. Higdon. Scaling Linux to the Extreme From 64 to 512 Processors. In *Linux Symposium*, July 2004.
19. T.-Y. Chiang, S. J. Souri, C. O. Chui, and K. C. Saraswat. Thermal analysis of heterogeneous 3-D ICs with various integration scenario. In *IEDM Technical Digest*, pp. 681–684, December 2001.
20. L. T. Clark, E. J. Hoffman, J. Miller, M. Biyani, Y. Liao, S. Strazdus, M. Morrow, K. E. Verlarde, and M. A. Yarch. An embedded 32-b microprocessor core for low-power and high-performance applications. *IEEE Journal of Solid State Circuits*, 36(11):1599–1608, November 2001.
21. E. L. Congduc. Packet classification in the NIC for improved SMP-based internet servers. In *Proceedings of International Conference on Networking*, February 2004.
22. W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon. Demystifying 3D ICs: The pros and cons of going vertical. *IEEE Design & Test of Computers*, 22(6):498–510, 2005.
23. M. J. Flynn and P. Hung. Computer architecture and technology: Some thoughts on the road ahead. In *Proceedings on International Conference on Engineering of Reconfigurable Systems and Algorithms*, pp. 3–16, 2004.
24. M. Ghosh and H.-H. S. Lee. Smart Refresh: An Enhanced Memory Controller Design for Reducing Energy in Converntional and 3D Die-Stacked DRAMs. In *International Symposuim on Microarchitecture*, December 2007.
25. B. Goplen and S. S. Sapatnekar. Thermal via placement in 3D ICs. In *Proceedings of International Symposium on Physical Design*, pp. 167–174, April 2005.
26. S. Gupta, M. Hilbert, S. Hong, and R. Patti. Techniques for producing 3D ICs with high-density interconnect. www.tezzaron.com/about/papers/ieee_vmic_2004_finalsecure.pdf.
27. R. Ho and M. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4), April 2001.
28. W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam. Compact thermal modeling for temperature-aware design. In *Proceedings Design Automation Conference*, June 2004.
29. T. Kgil. *Architecting Energy Efficient Servers*. PhD thesis, University of Michigan, 2007.
30. T. Kgil and T. Mudge. FlashCache: a NAND flash memory file cache for low power web servers. In *Proceedings of International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, October 2006.
31. T. Kgil, D. Roberts, and T. Mudge. Improving NAND flash based disk caches. In *Proceedings of International Symposium on Computer Architecture*, June 2008.
32. T. Kgil, A. Saidi, N. Binkert, S. Reinhardt, K. Flautner, and T. Mudge. PicoServer: Using 3D stacking technology to build energy efficient servers. *ACM Journal on Emerging Technologies in Computing Systems*, 2009.

33. M. G. Khatib, B. J. van der Zwaag, P. Hartel, and G. J. M. Smit. Interposing flash between disk and DRAM to save energy for streaming workloads. In *ESTIMedia*, 2007.
34. K. Kim and J. Choi. Future outlook of NAND flash technology for 40 nm node and beyond. In *Workshop on Non-Volatile Semiconductor Memory*, pp. 9–11, February 2006.
35. I. Koltsidas and S. D. Viglas. Flashing up the storage layer. In *VLDB*, August 2008.
36. P. Kongetira, K. Aingaran, and K. Olukotun. Niagara: A 32-way multithreaded Sparc processor. *IEEE Micro*, 25(2):21–29, March 2005.
37. M. Koyanagi. Different approaches to 3D chips. http://asia.stanford.edu/events/Spring05/slides/051205-Koyanagi.pdf.
38. S. R. Kunkel, R. J. Eickemeyer, M. H. Lipasti, T. J. Mullins, B. O'Krafka, H. Rosenberg, S. P. VanderWiel, P. L. Vitale, and L. D. Whitley. A performance methodology for commercial servers. *IBM Journal of Research and Development*, 44(6):851–872, 2000.
39. J. Laudon. Performance/watt: the new server focus. *SIGARCH Computer Architecture News*, 33(4):5–13, 2005.
40. K. Lee, T. Nakamura, T. Ono, Y. Yamada, T. Mizukusa, H. Hashimoto, K. Park, H. Kurino, and M. Koyanagi. Three-dimensional shared memory fabricated using wafer stacking technology. In *IEDM Technical Digest*, pp. 165–168, December 2000.
41. K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt. Understanding and designing new server architectures for emerging warehouse-computing environments. In *Proceedings of International Symposium on Computer Architecture*, June 2008.
42. J.-H. Lin, Y.-H. Chang, J.-W. Hsieh, T.-W. Kuo, and C.-C. Yang. A NOR emulation strategy over NAND flash memory. In *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, August 2007.
43. G. L. Loi, B. Agrawal, N. Srivastava, S.-C. Lin, T. Sherwood, and K. Banerjee. A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy. In *Proceedings Design Automation Conference*, June 2006.
44. J. Lu. Wafer-level 3D hyper-integration technology platform. www.rpi.edu/ luj/RPI_3D_Research_0504.pdf.
45. G. MacGillivray. Process vs. density in DRAMs. http://www.eetasia.com/ARTICLES/2005SEP/B/2005SEP01_STOR_TA.pdf.
46. D. A. Maltz and P. Bhagwat. TCP splicing for application layer proxy performance. Research Report RC 21139, IBM, March 1998.
47. R. E. Matick and S. E. Schuster. Logic-based eDRAM: origins and rationale for use. *IBM Journal of Research and Development*, 49(1):145–165, January 2005.
48. T. Ohsawa, K. Fujita, K. Hatsuda, T. Higashi, T. Shino, Y. Minami, H. Nakajima, M. Morikado, K. Inoh, T. Hamamoto, S. Watanabe, S. Fujii, and T. Furuyama. Design of a 128-Mb SOI DRAM using the floating body cell (FBC). *IEEE Journal of Solid State Circuits*, 41(1), January 2006.
49. C. Park, J.-U. Kang, S.-Y. Park, and J.-S. Kim. Energy-aware demand paging on NAND flash-based embedded storages. In *ISLPED*, pp. 338–343, 2004.
50. A. Rahman and R. Reif. System-level performance evaluation of three-dimensional integrated circuits. *IEEE Transactions on VLSI*, 8(6):671–678, December 2000.
51. F. Ricci, L. T. Clark, T. Beatty, W. Yu, A. Bashmakov, S. Demmons, E. Fox, J. Miller, M. Biyani, and J. Haigh. A 1.5 GHz 90 nm embedded microprocessor core. In *Proceedings of the IEEE Symposium on VLSI Circuits*, pp. 12–15, June 2005.
52. J. Scaramella. Enabling technologies for power and cooling. http://h71028.www7.hp.com/enterprise/downloads/Thermal_Logic.pdf.
53. J. Schutz and C. Webb. A scalable X86 CPU design for 90 nm process. In *Proceedings of IEEE International Solid-State Circuits Conference*, February 2004.
54. M. Shah, J. Barreh, J. Brooks, R. Golla, G. Grohoski, N. Gura, R. Hetherington, P. Jordan, M. Luttrell, C. Olson, B. Saha, D. Sheahan, L. Spracklen, and A. Wynn. UltraSPARC T2: A highly-threaded, power-efficient, SPARC SOC. In *Asian Solid-State Circuirts Conference*, November 2007.

55. J. Truong. Evolution of network memory. http://www.jedex.org/images/pdf/jack_troung_samsung.pdf.
56. D. Wendell, J. Lin, P. Kaushik, S. Seshadri, A. Wang, V. Sundararaman, P. Wang, H. McIntyre, S. Kim, W. Hsu, H. Park, G. Levinsky, J. Lu, M. Chirania, R. Heald, and P. Lazar. A 4 MB on-chip l2 cache for a 90 nm 1.6 GHz 64b SPARC microprocessor. In *Proceedings of IEEE International Solid-State Circuits Conference*, February 2004.
57. L. Xue, C. C. Liu, H.-S. Kim, S. Kim, and S. Tiwari. Three-dimensional integration: Technology, use, and issues for mixed-signal applications. *IEEE Transactions on Electron Devices*, 50:601–609, May 2003.

第 10 章　系统级三维（3D）集成电路成本分析与设计探索

Xiangyu Dong　Yuan Xie

目前大部分 3D IC 都集中在如何利用其在性能、功耗、更小外形和异构集成方面的优势。然而，当设计策略被确定时，所有的优势最终将转化为成本节约。因此，早期设计阶段的系统级成本分析是必要的，这有助于决定 3D 集成方案是否被采纳。在本章中，我们将讨论在早期设计阶段 3D 芯片设计的评估方法。我们还描述一个成本分析模型来研究 3D IC 成本的影响因素并寻求 3D IC 设计相关的成本问题。

10.1　介绍

目前大部分 3D IC 都集中在如何利用其在性能、功耗、更小外形和异构集成方面的优势。例如，7.9 节已经说明了 3D 设计的这些好处。但是，当涉及采纳这样的新兴技术作为主流设计方法时，这一切都归结于 3D 集成的成本问题。当设计策略被决定[18]时，3D 集成电路的所有优点最终转化为成本的节省。例如，设计人员可能会问自己一些问题，如难道 3D IC 设计的所有优点都会带来更大的成本？例如，3D 键合带来了额外的工艺成本，而且通过硅通孔（TSV）可能会增加总的芯片面积，这对成本有负面影响。然而，相比于 2D 电路，在 3D 集成电路中，更小的芯片尺寸可能会有更高的产量并且降低成本。

如何使 3D 集成电路保持成本效益？比如，重新设计一个小芯片可能无法获得来自 3D 集成电路高产量的成本优势。此外，如果实施芯片 3D 化，如何确定 3D 集成电路的层数以保证成本效益，并且应该使用晶圆到晶圆还是芯片对晶圆的堆叠[20]。

是否有设计方案能补偿 3D 集成电路额外的成本？例如，相比于每个 2D 芯片，在 3D 集成电路中，采用 TSV 技术的全局互连会减少所需的金属层数。此外，3D 异构集成通孔也可以帮助降低成本。

在早期设计阶段，3D 集成电路成本分析的关键是要回答上述问题，并且有助于做出 3D 集成电路是否应该被使用的决定并有助于判断哪种设计方案应该被采纳（如层数和键合方法）。在未来，广泛应用新兴的 3D IC 设计的关键是成本效益，并且 3D 集成电路的成本分析要与 3D IC 设计和 3D IC 工艺紧密相连。

本章中，我们首先描述 3D 芯片早期设计阶段（见 10.2 节）的设计评估方法，

并提出成本分析模型来研究 3D 集成电路（见 10.3 节）的成本。使用设计评估方法和 3D 成本分析模型，我们比较了 2D 和 3D 设计之间估算的成本，并调查了各种因素对成本的影响，以及 3D 集成带来的可能降低成本的方式。成本驱动的 3D IC 设计流程也被提出，它指导 3D IC 的设计向具有成本效益的方向发展（见 10.4 节）。

10.2　三维集成电路的早期设计评估

为了便于决定是否应该使用 3D 集成设计，从成本的角度来看，当没有太多详细的设计信息可利用时，有必要在初期设计阶段进行成本分析。IC 芯片的成本与芯片面积密切相关。对于 3D 芯片，TSV 技术可能产生额外的面积开销。然而，在 3D 集成电路的连线时可以使用较少金属层数量，进而有助于降低成本。

本节，我们将叙述在早期的设计阶段，当设计（例如估计在设计中的门数）仅有有限可用信息时，怎样评估芯片面积，评估易于灵活布线的金属层数，并评估 TSV 技术对芯片面积的影响。这种有利于 3D IC 成本分析的早期评估将在 10.3 节讨论。

10.2.1　“兰特规则”的初探

我们的早期设计估计是基于著名的兰特规则（Rent's Rule）[14]。兰特规则揭示信号终端的数目和内部的门数之间的对应关系。它基于现有设计观测到的经验化的结果，且可表示为

$$T = kN_g^p \tag{10-1}$$

式中，参数 k 和 p 为兰特规则的系数和指数；N_g 是门数；T 是终端信号的数目。

使用兰特规则，就能够进一步估计平均线长[11]和导线长度布局[9]。平均线长度可以通过下式给出

$$\overline{R}_m = \frac{2}{9}\,\frac{1-4^{p-1}}{1-N_g^{p-1}}\left(7\,\frac{N_g^{p-0.5}-1}{4^{p-0.5}-1} - \frac{1-N_g^{p-1.5}}{1-4^{p-1.5}}\right) \tag{10-2}$$

当 $p=0.5$ 时，所述表达可以使用洛必达法则[11]计算得出。

由兰特规则派生，导线长度分布函数 $i(l)$ 有以下几种形式：

区域 1：$1 \leqslant l \leqslant \sqrt{N_g}$

$$i(l) = \frac{\alpha k}{2}\Gamma\left(\frac{l^3}{3} - 2\sqrt{N_g}\,l^2 + 2N_g l\right)l^{2p-4} \tag{10-3}$$

区域 2：$\sqrt{N_g} \leqslant 1 \leqslant 2\sqrt{N_g}$

$$i(l) = \frac{\alpha k}{6}\Gamma\left(2\sqrt{N_g} - l\right)^3 l^{2p-4} \tag{10-4}$$

式中，l 是在各单元门间距的互连线长度；α 是片上接收终端的分数且与门平均扇出数相关（$f.o.$），如下：

$$\alpha=\frac{f.o.}{f.o.+1} \tag{10-5}$$

并且 Γ 是通过下式给出：

$$\Gamma=\frac{2N_g(1-N_g^{p-1})}{\left(-N_g^p\frac{1+2p-2^{2p-1}}{p(2p-1)(p-1)(2p-3)}-\frac{1}{6p}+\frac{2\sqrt{N_g}}{2p-1}-\frac{N_g}{p-1}\right)} \tag{10-6}$$

10.2.2　芯片面积和金属层估计

在早期的设计阶段，裸片面积可用门数代入函数进行估算：

$$A_{die}=N_gA_g \tag{10-7}$$

式中，N_g 是门的数量；A_g 是经验参数，显示面积和门数之间的比例关系。根据工业设计经验，我们在这项工作中假设 $A_g=3125\lambda^2$，其中 λ 为特定技术节点特征尺寸的一半。

布局所需金属层的数量取决于互连的复杂性。对于单一金属层的估计可以从平均线的长度得出[19]：

$$n_w=\frac{f.o.\ \overline{R}_m p_w}{e_w}\sqrt{\frac{N_g}{A_{die}}} \tag{10-8}$$

式中，$f.o.$ 指的是门平均扇出数；p_w 指的是线间距；e_w 指的是金属层的利用效率；$\overline{R}_m$ 指的是平均线长，由式（10-2）所确定；n_w 指的是金属层数。这种简化的模型基于各金属层具有同样的利用效率，并且线宽度相同的假设[19]。然而，这种假设可能无法在实际设计中成立[13]。此外，该模型中不包括 TSV 造成额外面积的影响，当 3D 集成电路的复杂性增加时，这种影响将造成非常严重的后果。

为了提高对金属层布线数量的估计的准确性，我们提出了一个新的 3D 布局布线模型，它基于金属线长分布，而不是平均线长的简单估计。该模型的基本思想解释如下：

用表达式估计每个金属层可用的布线区，即

$$K_i=\frac{A_{die}\eta i-2A_V(N_g f.o.-I(l_i))}{w_i} \tag{10-9}$$

式中，η_i 是利用效率；w_i 是线间距；A_v 是每个通孔的阻值；函数 $I(l)$ 是导线长度分布函数 $i(l)$ 的积分，在式（10-4）中已表达。

假设较短的互连线都在下部金属层布线。从金属 1 开始，我们将尽可能多的互连在当前金属层上布线，直到可用的布线区域被用完。布线时每个金属层的互连可以被表示为

$$\chi L(l_i)-\chi L(l_{i-1})\leqslant K_i \tag{10-10}$$

式中，$\chi=4/(f.o.+3)$ 是导线对同一网络互连线之间共享的一个因子[9,8]。函数 $L(l)$ 是 $i(l)$ 的一阶矩阵。

对每个金属层实行自下而上的方式重复计算，直到所有的互连布线正确。

通过应用上面介绍的估算方法，在早期设计阶段我们仅有作为输入的门数量时，可以预测芯片面积和金属层数。图 10-1 示出了估计面积和基于 65nm 工艺不同门数量级的电路金属层数的例子。

图 10-1 显示了一个重要含义，即在这种情况下 3D IC 成本会降低：当一个大的 2D 芯片被划分成多个较小的面积 3D 芯片堆叠时，每个较小的 3D 芯片需要的金属层的数量较少，可满足该互连布线的要求。这种金属层数的减少可抵消 3D 芯片堆叠造成的额外成本。

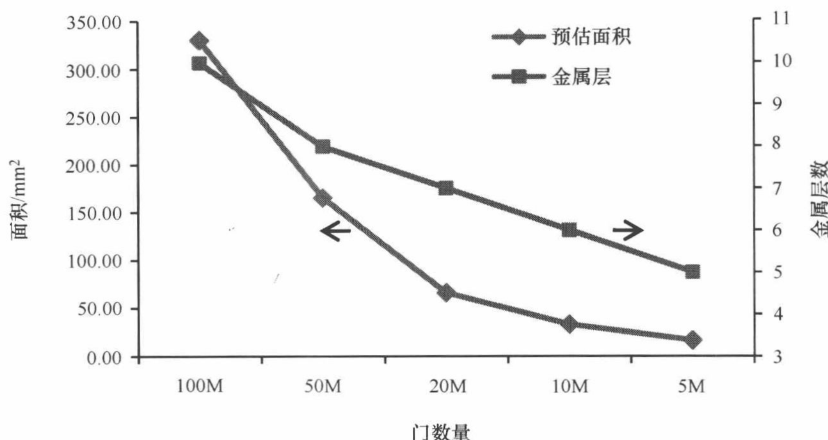


图 10-1　早期设计时的芯片面积和金属层数估计（65nm 工艺）。（目前这评估方法与国家最先进的微处理器设计紧密相关。例如，Sun SPARC T2[16] 包含 5 亿个晶体管（相当于 125 亿门），342mm² 面积和 11 个金属层）

10.2.3　TSV 技术的影响

在 3D 堆叠的成本分析中，硅通孔（TSV）的使用对其成本的影响是双重的：

在 3D 集成电路中，在堆叠裸片之间可使用一些全局性的互连 TSV 技术。这将会导致线长减小，并使得每个较小的芯片的金属层有减少的可能。

另一方面，3D 堆叠中的硅通孔（TSV）可能会增加总的芯片面积，是因为硅片上的 TSV 通孔可能不被用于器件连接或 2D 金属层连接（基于当前的 TSV 技术，其直径为 0.2 ~ 10μm[15]）。

因此，评估硅通孔（TSV）的数量对芯片面积增加的影响是很重要的。

为了预测一确定区域模式所需的硅通孔（TSV）的数量，互连（X）和门（N_g）之间的关系可用以下关系表示（兰特规则的推导[11]）：

$$X = \alpha k N_g (1 - N_g^{p-1}) \tag{10-11}$$

如图 10-2 所示，硅通孔（TSV）的数目可以由下式估算：

$$X_{TSV} = \alpha k_{1,2}(N_1 + N_2)(1 - (N_1 + N_2)^{p_{1,2}-1} - \alpha k_1 N_1(1 - N_1^{p_1-1})\alpha k_2 N_2(1 - N_2^{p_2-1}) \tag{10-12}$$

式中，$k_{1,2}$和$p_{1,2}$相当于兰特规则的系数和指数。由硅通孔（TSV）所引起的额外面积开销可以按如下方式建模：

$$A_{3D} = A_{die} + N_{TSV/die}A_{TSV} \tag{10-13}$$

式中，A_{die}由芯片面积估算；$N_{TSV/die}$为每个芯片上硅通孔（TSV）的数量；A_{TSV}是硅通孔（TSV）的大小；A_{3D}是3D芯片部分最终面积。

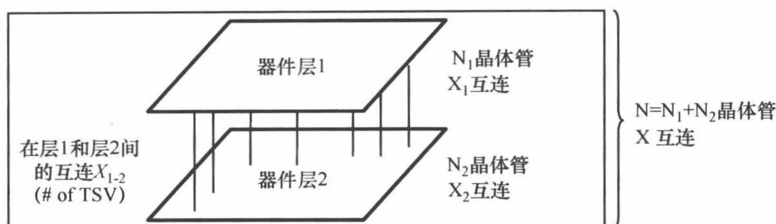


图 10-2　如何估算硅通孔（TSV）数目的基本思路

10.3　三维（3D）成本模型

三维集成电路包含由传统制造工艺堆叠多个芯片而成的芯片。有几种不同的方式将分立的芯片堆叠在一起[10]，其中以 TSV 为基础的方法是最有效的。除了传统的 2D 工艺，3D 集成电路需要额外的制造步骤，例如激光钻孔或蚀刻形成硅通孔（TSV），晶圆减薄和晶圆键合。

我们将 3D IC 制作过程的每一步成本进行模拟，并把成本分析分为芯片成本模型和三维键合成本模型，如图 10-3 所示。

晶圆成本模型。芯片成本模型的关键因素是芯片的面积。如果假设晶圆成本、产量和缺陷密度是常数，那么对使用特定技术节点的特定代工厂，芯片区域的影响可通过如下的两个表达式[17]表示：

$$N_{die} = \frac{\pi \times (\Phi_{wafer}/2)^2}{A_{die}} - \frac{\pi \times \Phi_{wafer}}{\sqrt{2 \times A_{die}}} \tag{10-14}$$

$$Y_{die} = Y_{wafer} \times \frac{(1 - e^{-2A_{die}D_o})}{2A_{die}D_o} \tag{10-15}$$

式中，N_{die}是每个晶圆芯片的数量；Φ_{wafer}是晶圆的直径；Y_{die}和Y_{wafer}是芯片和晶圆的产率；是D_o为晶圆的缺陷密度。

我们从不同晶圆代工厂得到的晶圆成本模型包括材料成本、人工成本、代工利润率、标线的数量、每个掩膜版的成本，以及其他杂项费用[3]。图 10-4 展示出 90nm、65nm、45nm 工艺分别用 9 层或 10 层的金属和 3 个不同晶圆代工厂所对应的晶圆成本。

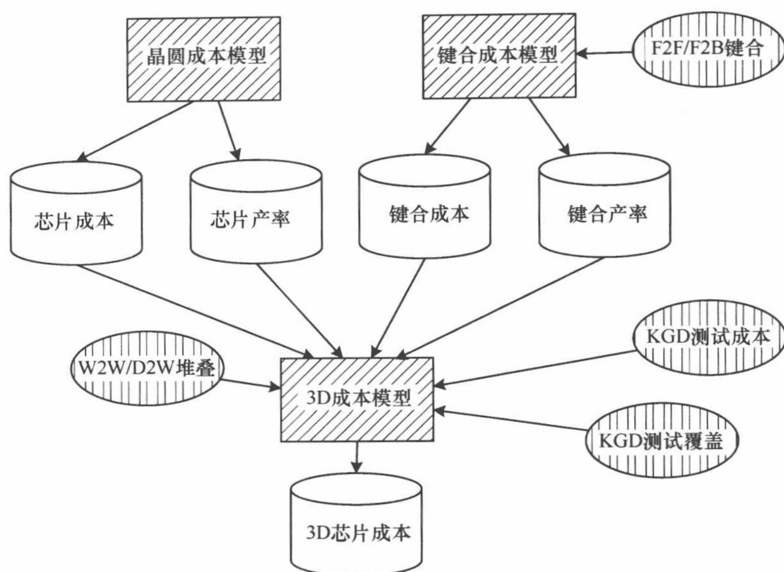


图 10-3　提出的 3D 成本模型概况图

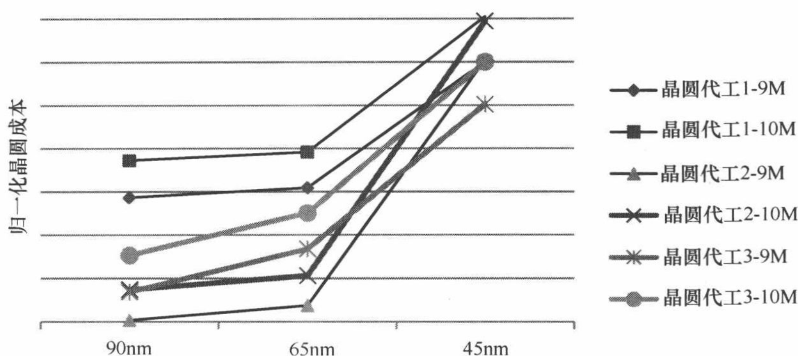


图 10-4　一组由晶圆成本模型计算的数据。由不同工艺、不同金属层数、不同晶圆代工厂以及其他一些因素对晶圆成本产生的变化

三维键合成本。第 2 章中描述了各种三维键合方法。3D 集成电路需要额外的步骤，即 TSV 形成、减薄和键合。在这项工作中，我们模拟了两种方法来实现 3D TSV 技术：激光钻孔和蚀刻。激光钻孔只适合少数的 TSV（数百至数千），而刻蚀适合于大量的 TSV。TSV（硅通孔）刻蚀过程类似于形成金属层之间的通孔，但正如它名字所暗示的，TSV 是“通过硅”。用于 TSV 蚀刻的两种方法：①TSV-first，TSV 可以在 2D 芯片制造工艺中形成，在后端 BEOL（back-end-of-line）工艺之前，这种方法被称为 TSV-first，如图 10-5a 所示；②TSV-last，TSV 也可以在 2D 工艺完成之后形成，在 BEOL（back-end-of-line）之后，这种方法被称为 TSV-last，如

图 10-5b 所示。我们的 3D 键合成本模型是基于行业合作伙伴提供的 3D 工艺，并假设每个 3D 工艺步骤的成品率达 99%。

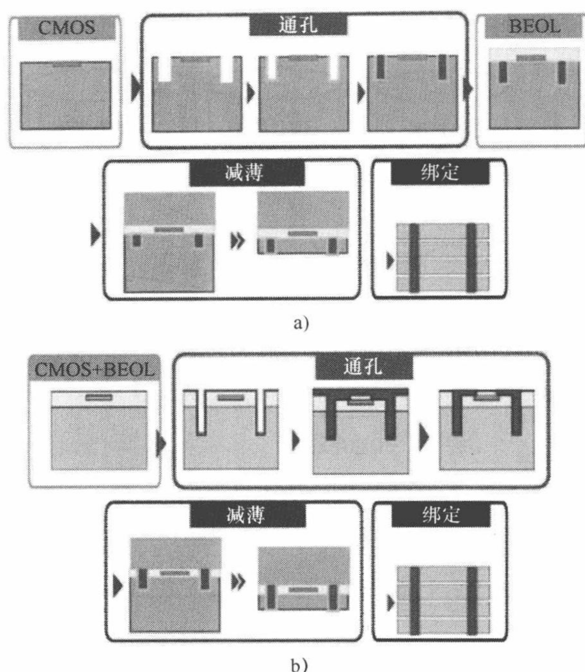


图 10-5　制备 3D 集成电路的步骤：a）TSV 是 BEOL 过程之前形成，从而 TSV 只能通过在硅衬底打孔而不是金属层；b）TSV 的形成是在 BEOL 工艺之后，从而 TSV 的通孔不仅在硅衬底，也在金属层

整个 3D 成本模型。除了晶圆成本模型和键合成本模型，整个 3D 成本模型还取决于一些设计方案，诸如芯片到晶圆/晶圆到晶圆的键合，面到面/面到背的键合，和已知良好的芯片的成本[18]。

对于 D2W 键合，封装前芯片成本的计算方法是

$$C_{D2W} = \frac{\sum_{i=1}^{N}(C_{die_i} + c_{KGDtest})/Y_{die_i} + (N-1)c_{bonding}}{Y_{bonding}^{N-1}} \tag{10-16}$$

对于 W2W 键合时，计算公式为

$$C_{W2W} = \frac{\sum_{i=1}^{N} c_{die_i} + (N-1)c_{bonding}}{(\prod_{i=1}^{N} Y_{die_i})Y_{bonding}^{N-1}} \tag{10-17}$$

为了支持多层键合，默认的键合方式是面对背。如果用面到面模式时，有多余一个不需要减薄加工的芯片，那么需要从总成本中减去该芯片的薄型化成本。

10.4　系统级三维 IC 设计探索

基于早期设计估计方法和前面章节中所描述的 3D 成本分析模型，我们使用 IBM 通用工艺平台成本模型在系统级执行一系列设计分析，调查不同设计方案对 3D IC 成本的影响，并且从成本的角度归纳了三维集成电路设计的几条规则。

10.4.1　评估 TSV 对芯片面积的影响

正如 10.2 节所说，在 3D IC 上构建的 TSV 不仅会带来额外的工艺成本，而且还导致面积成本。面积成本会影响芯片成品率和晶圆利用率。基于式（10-12），并根据 Bakoglu 的研究[5]设置指数参数 $P=0.63$ 和系数参数 $k=1.4$。我们进一步假设 3D 层数为 N，并且所有的门被均匀地划分为 N 层。我们选择的 TSV 的间距尺寸为 8μm。使用早期的设计估算，预测 65nm 工艺下对 TSV 的影响，如图 10-6 所示。

与式（10-13）相符合，TSV 面积成本随 3D 层数的增加而增加。对于 4 层的小规模设计（500 万门），其成本高达 10%。然而，当足够大的大规模设计（2 亿门）且 3D 集成仅堆叠两层芯片，TSV 面积成本通常低于 2%。总的来说，对于大的设计，TSV 的面积成本是可以接受的。

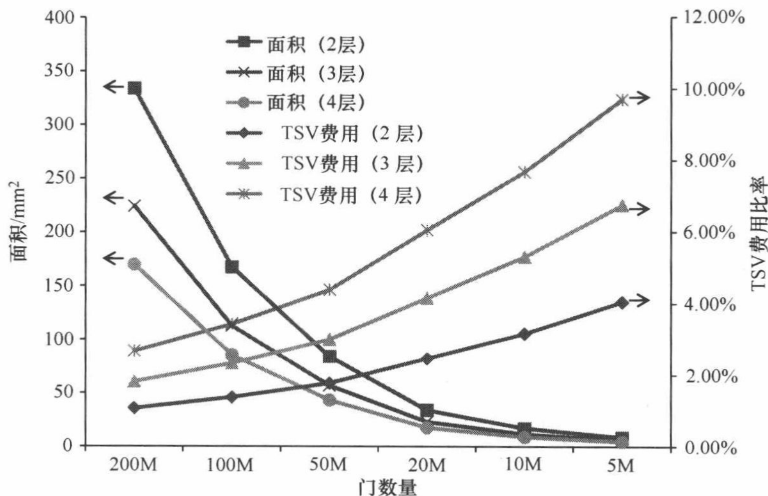


图 10-6　面积和 TSV 占总面积的百分比：对于小规模设计，TSV 面积成本接近到 10%，但对于较大规模设计，TSV 面积成本小于 4%

10.4.2　三维（3D）IC 中减少金属层的潜力

从理论上说，当门的数目被均匀地分布到多个 3D 堆叠芯片中时，每个较小芯片上的总线长度等于大规模 2D 芯片的整体线长除以芯片个数。此外，正如 10.2

节中所说，由于 TSV 的存在，总线长度随着 3D 芯片层数的增加而降低。把这两个因素一起比较，可知每层芯片的布线复杂度比 2D 基线设计的复杂度小得多。其结果就是使 3D 芯片堆叠中的每一个小芯片都能移除一个或两个金属层。

通过 10.2 节中讨论的 3D 芯片布线模型的评估，我们可以预测金属层数降低的影响，结果见表 10-1。

表 10-1 每个芯片所需的金属层的数量（65nm 微处理器）

门数	1 层 2D	2 层 3D	3 层 3D	4 层 3D
5M	5	5	5	4
10M	6	5	5	5
20M	7	6	5	5
50M	8	7	7	6
100M	10	8	7	7
200M	12	10	9	8

尽管结果表明在相对较小规模的设计中，几乎不能降低金属层数（如 500 万门设计），但随着设计复杂度的增长，金属层数的减小变得越来越明显。例如，当设计大规模 2D（即 2 亿门）时，通过均匀地将其划分为 2、3 或 4 个独立的芯片来进行 3D 堆叠，相应地可以使金属层数减少 2、3 或 4 层。

总而言之，在 3D 集成电路的设计中，相对于 2D 布线设计，有可能在每个小规模芯片中使用较少数量的金属层数。这种金属层数的减少可以抵消 3D 集成电路额外的键合成本。

10.4.3 键合工艺：D2W 或 W2W

在 3D 集成电路中，芯片到晶圆（D2W）和晶圆到晶圆（W2W）是两种不同的键合多芯片的方法[10]。10.3 节通过引入良品测试（KGD）讨论了这 2 种建模方法，D2W 键合可以达到更高的生产率。然而 W2W 在键合前并不需要任何测试，而以产量为代价，将易于具有高的吞吐量的裸片对准。因为 D2W 和 W2W 都有自己的优点和缺点，我们可以用 3D 的成本模型找出哪一个更适合 3D 集成技术。

图 10-7 展示出传统 2D 工艺、两层 D2W 键合和两层 W2W 键合的成本对比。可以观察到的是，在小规模设计情况下，虽然 W2W 的成本少于 D2W，但是 W2W 的成本依旧比 2D 工艺成本更高。这个现象可通过面积和产量之间的关系来解释，并可以由式（10-15）表达。对于 W2W 键合，每层芯片组件的产量由于面积的减小而有所增加，当所有不做叠前测试的芯片被堆叠在一起，整个芯片的成品率等于每层芯片组件成品率。因而 W2W 键合的 3D 芯片总成品率与 2D 芯片一样低。在考虑额外键合成本之后，便能合理的解释为什么 W2W 一直比传统 2D 花费的更多。

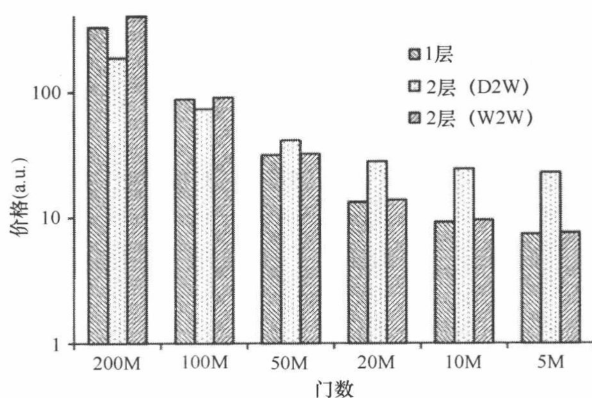


图 10-7　在 65nm 工艺下 2D、双层 D2W 和双层 W2W 的成本对比
（考虑了 TSV 面积开销和金属层的减少）

总之，从产品率的角度来看，基于我们的晶圆成本模型和 3D 键合成本模型来看，芯片到晶圆（D2W）比晶圆到晶圆（W2W）更具有成本优势。

10.4.4　成本与三维层数

基于设计早期阶段的评估方法可以预测 3D IC 芯片面积、3D 的 TSV 影响和金属层数减少的影响，我们可以进一步使用这些设计相关的参数作为输入，输入到 10.3 节所提出的 3D IC 成本模型，并评估每个 3D 设计方案的成本。

首先，我们选择 IBM 公司的 65nm 通用平台模型，并把 2D 基线设计成本分别与其相应的 2、3 和 4 个芯片堆叠在一起的 3D 芯片作对比。图 10-8 显示出成本评估是基于 2D 布线设计均匀划分的假设。

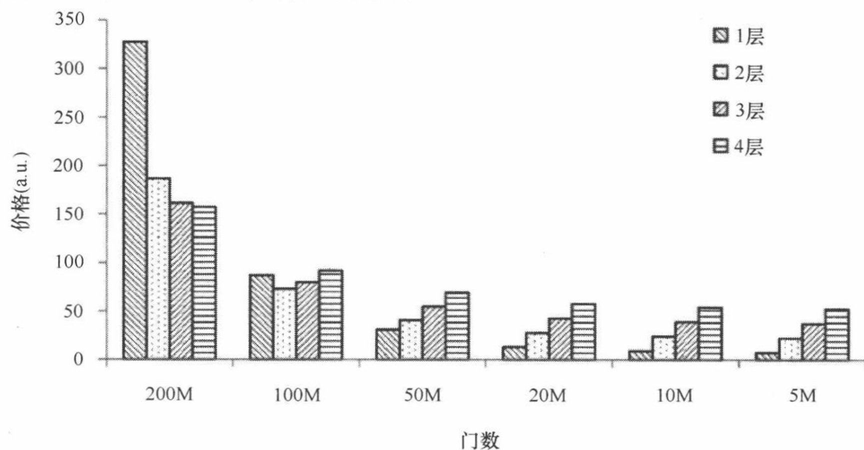


图 10-8　不同层数的 3D 集成电路在 65nm 制程下的成本
（考虑了 TSV 面积开销和金属层的减少）

从图 10-8 中可以观察到，由于芯片尺寸和芯片产量之间的指数关系，成本随着芯片尺寸的增长急剧增加。因为当芯片面积较大时，产量对芯片面积的变化会十分敏感，所以对于将芯片分割成多个来说，大规模 2D 芯片设计会比小规模 2D 芯片设计更有可能减少整体成本。

另外需要注意的是，3D 芯片的最佳层数（就成本而言）会有所不同，它取决于设计的大小。例如，对于 2 亿门最具成本效益的方式就是做一个 4 层 3D 分割的 3D 组合；但对于 1 亿门的设计，最具有成本效益的方法是使用 2 层 3D 集成；当最初的 2D 设计相当小（ <5000 万）时，传统的 2D 工艺依旧是最便宜的，这是因为此时 3D 键合成本开始处于主导地位且小规模 3D 组合可以提高产量。

在 IBM 公司的通用平台技术模型上使用不同工艺重复实验，我们估计出一组边界值，显示出两层的 3D 堆叠开始比传统 2D 工艺更具成本效益的边界。该数据列于表 10-2。如果我们把门的数量转换为芯片尺寸，两层 3D 工艺的应用拐点为约 250mm^2。两层以上芯片堆叠的应用拐点可能会更大。

表 10-2 3D 制造工艺的应用拐点

工艺（IBM 普通平台）	45nm	65nm	90nm	130nm
拐点（门数）	143M	76M	40M	21M

总结一下，大规模 3D 集成电路具有成本效益，而非小规模设计；3D 集成电路的层数（从成本的角度来看）的最佳数目随门数增加而增加。

10.4.5 异构堆叠

所有上述讨论都集中在同构堆叠。但是 3D 集成电路最大的优点之一是它支持异构堆叠，因为其可以由不同类型的部件制成。

用目前高性能微处理器作为例子，硅面积上的很大一部分是被片上 SRAM 或 DRAM 占据，并且非易失性存储器可以被集成于片上存储器[12]。然而，对于不同的模块来说，其相应的制造工艺也是不同的。例如，当传统常规 CMOS 逻辑电路需要 1-多晶硅-9-铜-1-铝互连层时，DRAM 模块需要的是 7-多晶硅-3-铜，而闪存模块则需要的是 4-多晶硅-1-钨-2-铝。其结果是异构集成将显著地增加成本。比如，Intel 公司的研究表明大规模 2D SoC 异构集成会将芯片成本提高 3 倍[7]。

分离异构制造技术和 3D 集成堆叠组件会对这些系统产生成本效益。在这里，我们采用 OpenSPARC T2[16] 来研究。最初 2D 的 OpenSPARC T2 芯片有 342mm^2 的面积，它采用 TI 65nm 制造工艺和 11 层金属数。大约一半的芯片面积将会给片上 SRAM 缓存使用。对于这种微处理器，使用 3D 集成的一种办法就是将所有的 SRAM 模块从一个芯片上分开，并且所剩其他模块集成在其他的芯片上，这与近期 Intel 公司发布的 80 核万亿级芯片相似[1]。应用 10.2 节早期设计的评估方法，并选择用于 SRAM 的兰特参数 $p=0.12$，$k=6$，我们由此可估计出 SRAM 模块的金属层

数可减少到 5。并且我们可以进一步通过使用 3D IC 成本模型来估计芯片总成本。这些对比见图 10-9。

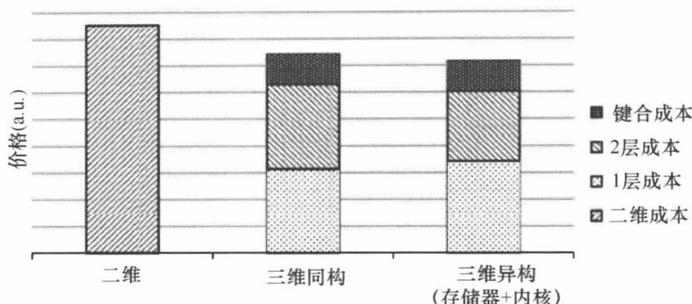


图 10-9　分别估算使用传统 2D、同构 3D 划分和异构 3D 划分制成的 OpenSPARC T2 的成本：制造该存储器和分立的核心部分可以进一步节省成本

总之，异构集成提供了额外的机会能够使 3D IC 设计的总成本降低。

10.5　成本驱动型的三维设计流程

在正式设计前，对 3D IC 成本分析进行上述讨论，并将早期设计的评估预测作为成本模型全部地输入。然而，如果在设计时期应用相同的成本分析方法，并使用真实的设计数据，如芯片面积、TSV 互连和金属互连作为成本模型的输入，这会使成本驱动型 3D IC 设计流程成为可能。图 10-10 展示了所提出的成本推动 3D IC 设计流程。设计流程中加入了集成 3D IC 成本分析模型，指导设计者优化 3D IC 设计并最终制造出低成本的产品。

这样的成本分析/减小的 EDA 流程由 3 组操作组成：设计相关操作、成本建模相关的操作和降低成本操作。

- 设计相关操作　包括 3D 划分、时序分析和布局布线，所有这些都是典型的 3D 芯片设计流程。这些操作会影响成本评估。例如，不同的划分策略可能导致不同的组件在一个芯片上有不同数目的 I/O 接口。布局布线确定每层的互连拓扑结构，并导致 3D 集成电路需要不同数目的硅通孔，这会影响键合费用。
- 成本模型操作　包括芯片面积评估（计算每层芯片面积）、晶圆成本模型（计算每个组合层的成本）、3D 键合成本模型（计算堆叠多层芯片的成本和制造 3D 通孔的成本）、堆叠成本模型、用于评估相关不同堆叠方案的选择成本。例如，芯片到晶圆（D2W）堆叠需要在顶层其他芯片堆叠前进行良品测试，并会产生额外的芯片测试费用；但它可以提高堆叠芯片的产量。这些模型在前面的章节中已经说明。
- 降低成本操作　包含可能的降低成本的办法。例如，一种方法被称为异构

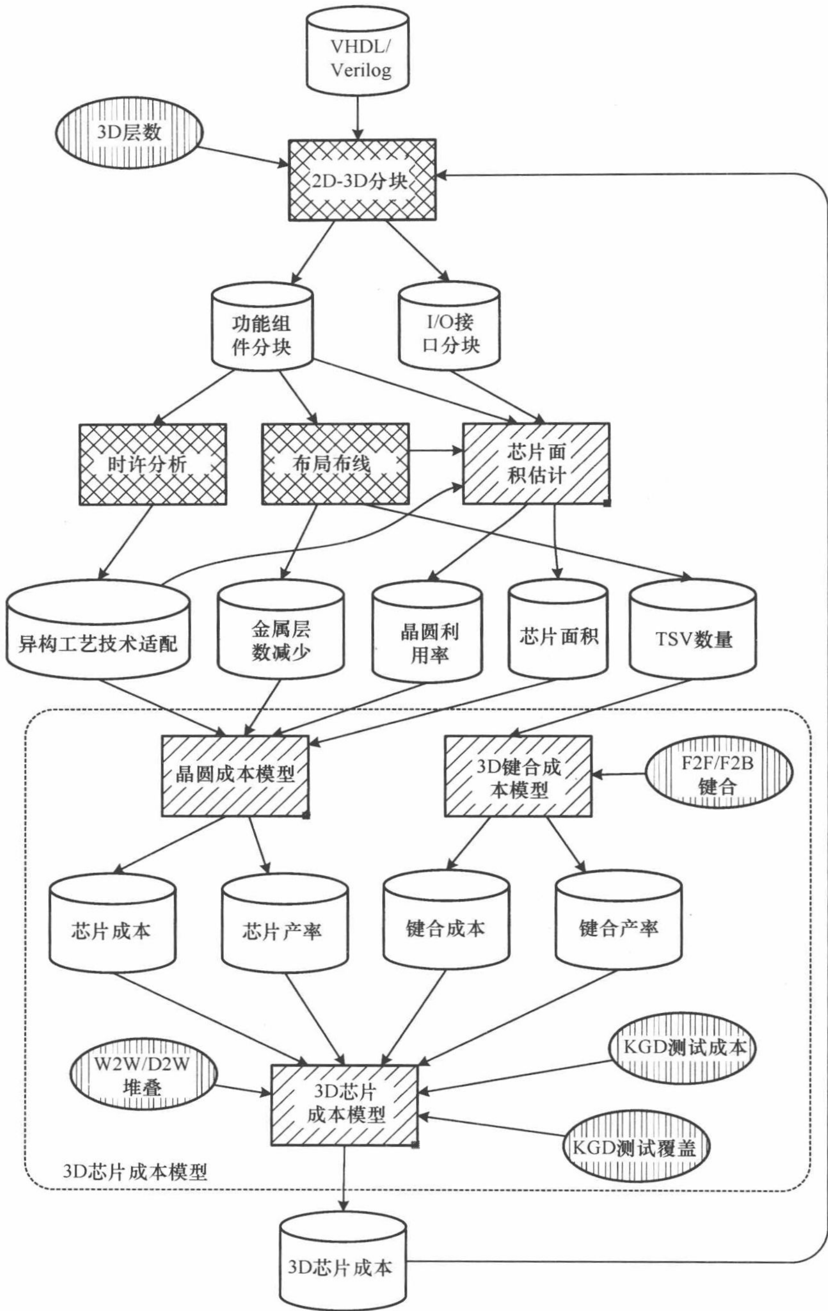


图 10-10 成本驱动型 3D IC 设计流程的方案

工艺技术堆叠：在被分区到芯片上的组件不是至关重要时，可用速度较慢（较便宜）的制造工艺（如0.18μm CMOS工艺）。当分区到芯片上的组件是至关重要的芯片时，要用更高级（快速但是昂贵）的制造技术（如65nm CMOS工艺）。第二种方法被称为金属层减少法：当设计从2D变到3D时，每个芯片自身使用更少的金属层数布线，这可以节省后端工艺成本。

用3D EDA设计流程作为一个独特而紧密的集成成本分析具有两方面的优势。首先，正如我们前面讨论的，许多设计决策（如划分和布局布线）会影响成本分析。用3D EDA流程做紧密耦合的成本分析会使结果评估成本更加准确。第二，除了考虑其他设计目标（如性能和功率），成本分析结果可以驱使3D EDA工具进行更具成本效益的优化。

10.5.1　案例分析：两层 OpenSPARC T1 三维处理器

我们使用Sun公司的OpenSPARC T1处理器[2]作为案例研究，以说明如何通过最后一节提到的节省成本的方法来抵消相关3D技术的额外制作成本。降低成本的结果可使一个3D芯片的总成本将会比相应的2D芯片低。

正如我们上节所提到的，有两个3D节省成本的主要方法：①金属层减少，在制造期间通过利用3D技术引入的三维布线减少金属层数；②异构工艺技术堆叠，使用较旧和较便宜的工艺点来制造划分非关键组件到指定层。

有两种划分方式可以来帮助3D堆叠降低成本：①粗度划分。所述的OpenSPARC T1处理器可以分成处理器内核和缓存单元块；在10.4.5节，我们可以看到这些分区能帮助降低成本，是因为从逻辑层分离存储器可以降低成本。②精度划分。此方法中，在单元层次[6]我们使用图10-10所提到的成本驱动流程来划分组件。随着这样的流程，我们将整个8核OpenSPARC T1处理器分成两层精度划分并保证时序要求不变。利用这种精度划分，有两种可行的办法来节省成本：

1）90nm与90nm堆叠。在此方法中，两个层均使用90nm技术。我们将这些单元精细分割成两层，使所得到层的面积彼此相等，并保持关键路径不变。从使用的90nm标准单元库的综合结果中，我们观察到在8核SPARC T1单个芯片的总面积约为10.63mm^2。用两层3D划分，单个芯片面积减小至7.18mm^2和7.03mm^2。根据我们的成本模型，相比于传统2D 146美元的成本，使用3D的成本是125美元。

2）90nm与130nm异构工艺技术堆叠。在此方法中，利用时序分析结果可以找出一组不在关键路径上且能移到较慢的层上的组件，对其在更慢的层上使用130nm标准单元库进行综合。基于该综合结果的成本分析，此成本可以进一步降低到121美元。

10.6 交互对称设计的三维掩膜版的重复使用

前面的章节中仅讨论了制造成本的建模，并不包含掩膜版成本。通常，每个用于堆叠的芯片需要一套特别的掩膜版，因此相比于对应的2D 芯片，掩膜版的成本使得3D 芯片的成本有所增加。掩膜版成本会随技术规模急剧增加，因此，一组3D IC 掩膜版成本将会对3D IC 芯片的最终成本产生显著影响。然而，由于缺乏掩膜版成本模型，以及掩膜版成本通常随特定应用变化，我们在本章所描述的3D 成本模型中不包括掩膜版的成本。

尽管在一般情况下，一个3D IC 芯片需要多套掩膜版，有些特殊的设计案例，如存储器堆叠，其允许所有存储器层重复使用一套掩膜版。另外，在第10.4.5 节中，我们已经表明，如果基于 SRAM 的二级缓存被堆叠在处理器内核顶层，则缓存模块的金属层数可减少到5 层，代替了11 层的处理器内核，这使得成本有实质性的降低，如图10-9 所示。另外，在10.4.2 节，3D 集成电路金属层数降低的潜力已被证实。这种金属层的减少不仅降低了生产成本，而且还有助于抵消因为3D 堆叠产生的额外掩膜版成本。

Alam 等人[4]最近提出了使用一种新颖的设计技术，即交互对称设计（RDS）。它能使在3D 堆叠裸片时的一套掩膜版可以被其他层重复使用。RDS 的思想可以用图10-11 所示的双核存储器堆叠例子进行说明。

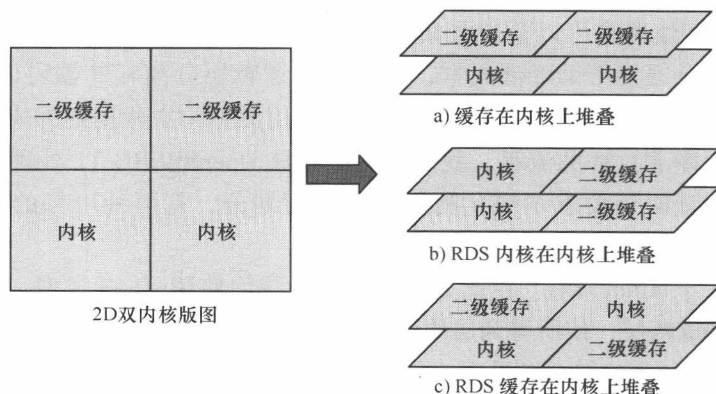


图10-11 实现3D 双核微处理器的3 种方法

如图10-11 所示，实现3D 双核微处理器可用的3 种不同方法：①对于双核来说将两层所有的二级缓存移到顶层核上。这与7.2 节（图7-2a）描述的方法相同。这种方法需要两套掩膜版：一个用于内核层和其他缓存层。②第二个方法是让内核处于内核上（缓存处于缓存上）的堆叠。这种方法能让单套掩膜版用于两层。然而，因为内核的功率密度通常要远远大于缓存，所以这样的堆叠方法会导致更高的

温度上升。③第三种方法是将第二种方法中的一个层旋转，以实现缓存处于内核之上（内核处于缓存之上）的堆叠。这种方法能在交互对称设计中重复使用掩膜版，同时减小热增加带来的影响。

10.7　结论

为了克服技术进步带来的障碍，对于未来的集成设计来说，3D 集成电路是具有吸引力的。然而，若要广泛使用 3D 集成电路，制造成本则是需要重要考虑的因素之一。在早期的设计阶段的系统级成本分析，决定在应用中是否使用 3D 集成电路是非常关键的。

为了便于系统级成本分析，我们研究在早期设计阶段对 3D IC 设计的评估方法并提出成本分析模型来研究对成本的影响。对于成本的分析，我们确定了在 3D IC 设计降低成本的机会，并对具有成本效益的设计提供了一些设计指引。我们的研究是与目前 3D IC 在其他设计目标（如性能和功耗分析）的研究相辅相成的。

致　谢

感谢来自半导体制造技术战略联盟（SEMATECH）的 Larry Smith 博士，来自 IBM 公司的 Mike Ignatowski 博士，来自高通（Qualcomm）公司的 Sam Gu 博士和来自微电子研究中心（IMEC）的 Pol Marchal 博士对这项研究给予有价值的讨论和指导。这项工作是在 NSF 的 CAREER 0643902 和 CCF 0702617 的部分资助，高通（Qualcomm）公司赠款，以及 IBM Faculty 的共同支持下完成的。

参考文献

1. http://techresearch.intel.com/articles/Tera-Scale/1421.htm. 2007.
2. http://www.opensparc.net/. 2008.
3. IC Cost Model, 2008 revision 0808a. In *IC Knowledge LLC*, 2008.
4. S. Alam, R. Jones, S. Pozder, and A. Jain. Die/wafer stacking with reciprocal design symmetry (rds) for mask reuse in three-dimensional (3D) integration technology. In *International Symposium on Quality Electronic Devices*, 2009.
5. H. B. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, Reading, MA, 1990.
6. K. Bernstein. New dimension in performance. *EDA Forum*, 3(2), 2006.
7. S. Borkar. 3D-Technology: a system perspective. In *International 3D-System Integration Conference*, 2008.
8. P. Chong and R. K. Brayton. Estimating and optimizing routing utilization in DSM design. In *Workshop System-Level Interconnect Prediction*, 1999.
9. J. A. Davis, V. K. De, and J. D. Meindl. A stochastic wire-length distribution for gigascale integration derivation and validaton. derivation and validation. *IEEE Transaction on Electron Devices*, 45(3):580–589, 1998.

10. W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon. Demystifying 3d ics: the pros and cons of going vertical. *IEEE Design and Test of Computers*, 22(6):498–510, 2005.
11. W. E. Donath. Placement and average interconnection lengths of computer logic. *IEEE Trans. on Circuits and Systems*, 26(4):272–277, 1979.
12. X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, and Y. Chen. Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement. In *Design Automation Conference*, 2008.
13. A. B. Kahng, S. Mantik, and D. Stroobandt. Toward accurate models of achievable routing. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 20(5):648–659, 2001.
14. B. S. Landman and R. L. Russo. On a pin versus block relationship for partitions of logic graphs. *IEEE Trans. on Computers*, C-20(12):1469–1479, 1971.
15. G. H. Loh, Y. Xie, and B. Black. Processor design in 3D die-stacking technologies. MICRO, 27(3):31–48, 2007.
16. S. C. Marc Tremblay. A third-generation 65 nm 16-core 32-thread plus 32-scout-thread CMT SPARC(R) Processor. In *International Solid State Circuit Conference*, pp. 82–83, 2008.
17. J. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits*. Prentice-Hall, Englewood Cliffs, NJ, 2003.
18. L. Smith, G. Smith, S. Hosali, and S. Arkalgud. 3D: it all comes down to cost. *Proceedings of RTI Conference of 3D Architecture for Semiconductors and Packaging*, 2007.
19. R. Weerasekera, L.-R. Zheng, D. Pamunuwa, and H. Tenhunen. Extending systems-on-chip to the third dimension: performance, cost and technological tradeoffs. In *ICCAD*, pp. 212–219, 2007.
20. Y. Xie, G. H. Loh, B. Black, and K. Bernstein. Design space exploration for 3D architectures. *J. Emerg. Technol. Comput. Syst.*, 2(2):65–103, 2006.

读者需求调查表

亲爱的读者朋友：

您好！为了提升我们图书出版工作的有效性，为您提供更好的图书产品和服务，我们进行此次关于读者需求的调研活动，恳请您在百忙之中予以协助，留下您宝贵的意见与建议！

个人信息

姓　　名：		出生年月：		学　　历：	
联系电话：		手　　机：		E-mail：	
工作单位：				职　　务：	
通讯地址：				邮　　编：	

1. 您感兴趣的科技类图书有哪些？

☐ 自动化技术　☐ 电工技术　☐ 电力技术　☐ 电子技术　☐ 仪器仪表　☐ 建筑电气

☐ 其他（　　）

以上各大类中您最关心的细分技术（如 PLC）是：（　　）

2. 您关注的图书类型有

☐ 技术手册　☐ 产品手册　☐ 基础入门　☐ 产品应用　☐ 产品设计　☐ 维修维护

☐ 技能培训　☐ 技能技巧　☐ 识图读图　☐ 技术原理　☐ 实操　☐ 应用软件

☐ 其他（　　）

3. 您最喜欢的图书叙述形式

☐ 问答型　☐ 论述型　☐ 实例型　☐ 图文对照　☐ 图表　☐ 其他（　　）

4. 您最喜欢的图书开本

☐ 口袋本　☐ 32 开　☐ B5　☐ 16 开　☐ 图册　☐ 其他（　　）

5. 图书信息获得渠道：

☐ 图书征订单　☐ 图书目录　☐ 书店查询　☐ 书店广告　☐ 网络书店　☐ 专业网站

☐ 专业杂志　☐ 专业报纸　☐ 专业会议　☐ 朋友介绍　☐ 其他（　　）

6. 购书途径

☐ 书店　☐ 网络　☐ 出版社　☐ 单位集中采购 ☐ 其他（　　）

7. 您认为图书的合理价位是（元/册）：

手册（　　）图册（　　）技术应用（　　）技能培训（　　）基础入门（　　）其他（　　）

8. 每年购书费用

☐ 100 元以下　☐ 101 ~ 200 元　☐ 201 ~ 300 元　☐ 300 元以上

9. 您是否有本专业的写作计划？

☐ 否　☐ 是（具体情况：　　　　）

非常感谢您对我们的支持，如果您还有什么问题欢迎和我们联系沟通！

地址：北京市西城区百万庄大街 22 号　机械工业出版社电工电子分社

邮编：100037

联系人：任鑫　联系电话：010-88379767　传真：010-68326336

电子邮箱：balance008@126.com（可来信索取本表电子版）

编著图书推荐表

姓名		出生年月		职称/职务		专业	
单位				E-mail			
通讯地址						邮政编码	
联系电话		研究方向及教学科目					
个人简历（毕业院校、专业、从事过的以及正在从事的项目、发表过的论文）							
您近期的写作计划有：							
您推荐的国外原版图书有：							
您认为目前市场上最缺乏的图书及类型有：							

地址：北京市西城区百万庄大街 22 号　机械工业出版社　电工电子分社
邮编：100037　网址：www. cmpbook. com
联系人：任鑫　电话：13811709139　010-68326336（传真）
E-mail：balance008@126. com（可来信索取本表电子版）

国际视野 科技前沿

本书是3D设计领域的综述，重点在于使3D技术被采纳的EDA工具和算法，实施架构和在未来的、潜在的3D系统设计。本书旨在为读者提供全面的认识，主要介绍了以下内容：

- 3D集成电路技术是一种有效的设计方法，使得芯片工业能够沿着性能提高的道路继续发展。
- 3D集成电路技术的工艺介绍。
- 3D集成电路技术面临的特殊的关于EDA的挑战，以及解决方法和最佳实践。
- 使用3D技术的优势。
- 架构和系统级设计问题。
- 3D集成电路设计的成本。

国际信息工程先进技术译丛

《3D集成电路设计——EDA、设计和微体系结构》
《ROF光载无线通信：从理论到前沿》
《全面详解LTE：MATLAB建模、仿真与实现》
《低速无线个域网：实现基于IEEE 802.15.4的无线传感器网络（原书第3版）》
《6LoWPAN：无线嵌入式物联网》
《虚拟网络——下一代互联网的多元化方法》
《Android系统安全与攻防》
《移动无线信道》（原书第2版）
《LTE-Advanced：面向IMT-Advanced的3GPP解决方案》
《认知无线电通信与组网：原理与应用》
《LTE/SAE网络部署实用指南》
《IP地址管理原理与实践》
《自组织网络：GSM，UMTS和LTE的自规划、自优化和自愈合》
《实现吉比特传输的60GHz无线通信技术》
《LTE自组织网络（SON）：高效的网络管理自动化》
《UMTS中的LTE：向LTE-Advanced演进》（原书第2版）
《UMTS中的WCDMA－HSPA演进及LTE》（原书第5版）
《UMTS中的LTE:基于OFDMA和SC-FDMA的无线接入》
《基于4G系统的移动服务技术》
《UMTS蜂窝系统的Q0S与Q0E管理》
《UMTS-HSDPA系统的TCP性能》
《基于射频工程的UMTS空中接口设计与网络运行》
《基于蜂窝系统的IMS—融合电信领域的V0IP演进》



机械工业出版社E视界



机械工业出版社微信公众号

上架指导 工业技术／集成电路



ISBN 978-7-111-52605-6 定价：79.00元